JULY/AUGUST 86
VOL.2 NO.5
$3.00

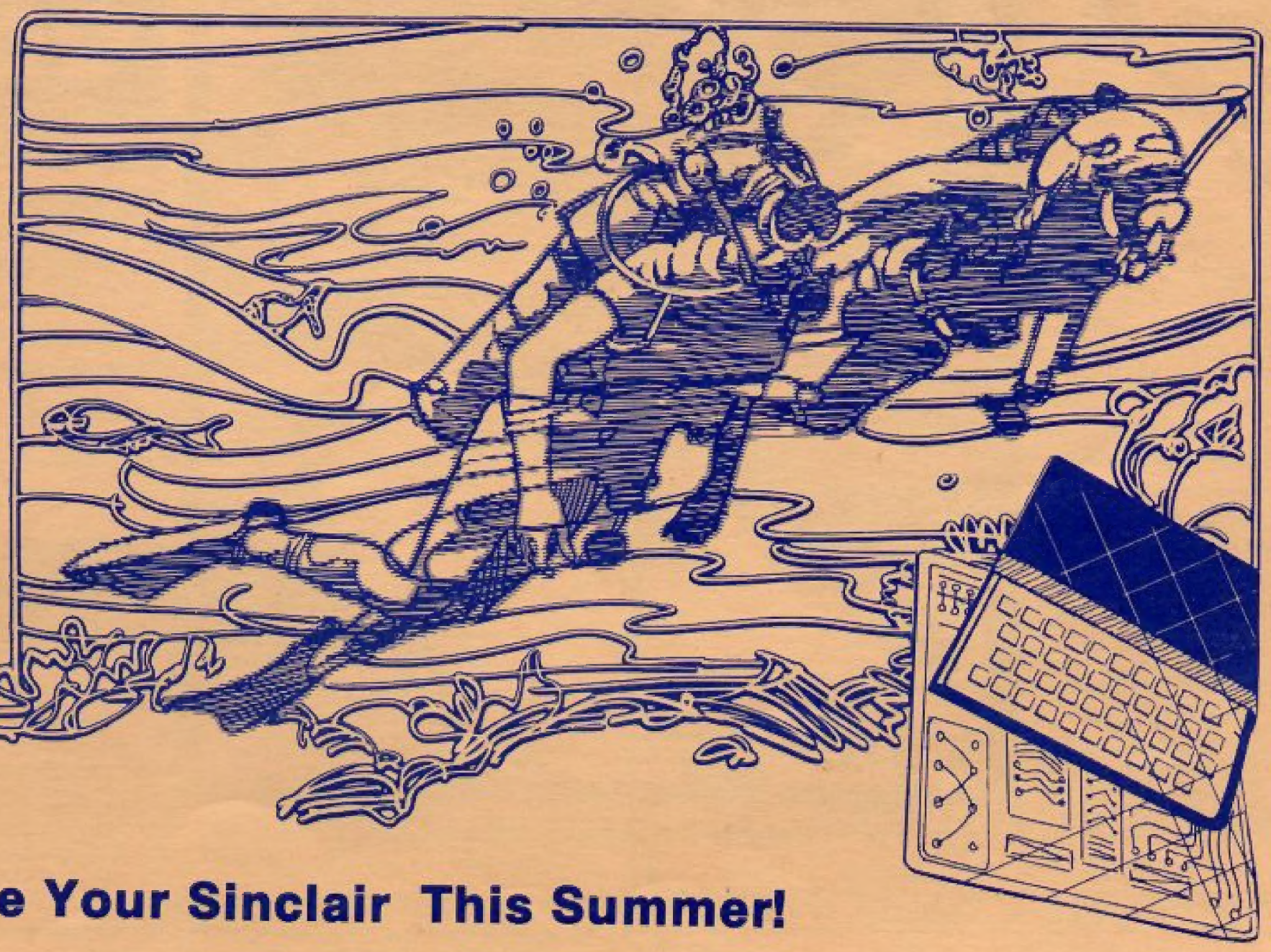
BULK RATE
U.S. POSTAGE
PAID
COLTON, OR 97017
PERMIT #51

FOR ALL TIMEX AND SINCLAIR COMPUTERS

time Designer™

MAGAZINE

T/S 1000 External Keyboard Buffer • QL Games
2068 Graphics • "The Mystery of the Missing 253"
And Lots More...



Explore Your Sinclair This Summer!

curry computer

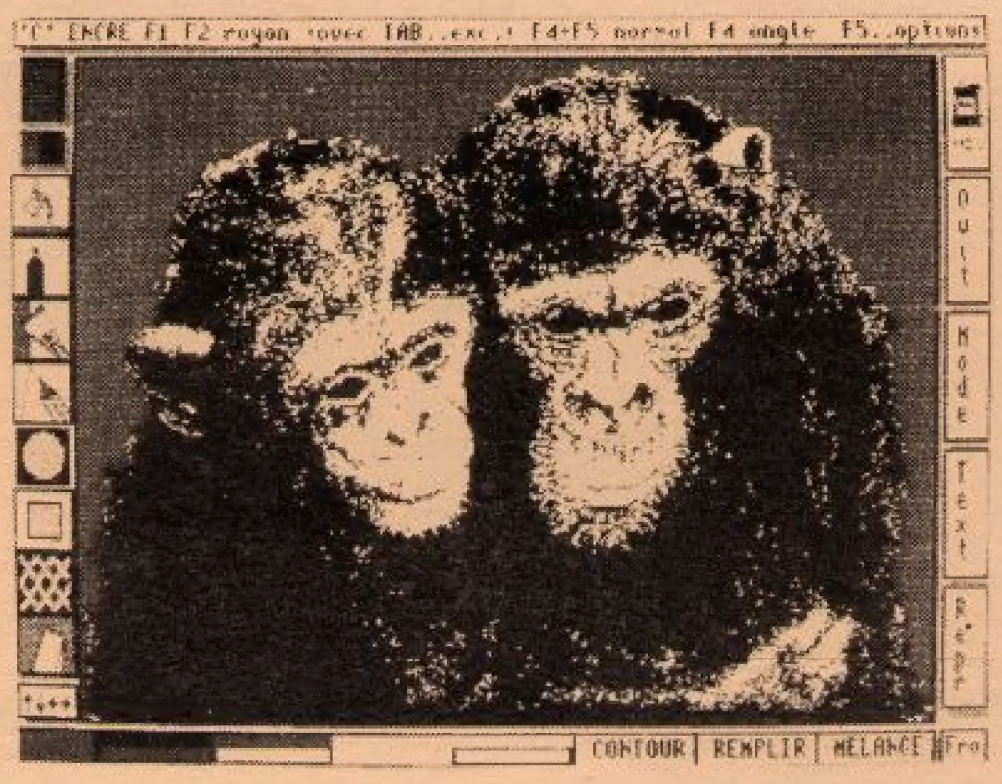
"We know the QL ... We've been appreciating it longer."

Curry Computer has been Appointed Sole U.S. Distributor for all Pyramide QL Products

Also from Pyramide and Curry Computer: Nucleon, Vroom, Othello and Mortville's Manor

QL-Peintre

$34.95



Available from Curry Computer or Your Local QL Dealer

"The best up and coming game for the QL: Sinclair User March 1986

PYRAMIDE

(RGB COLOR MONITOR ONLY)

SOFT FOR THE QL ONLY

For the first time, a game which uses the full power of the QL spatial effects, combined with fast animation. The object will leap out of the screen thanks to the

supplied with the game

TRADE INQUIRES WELCOME

P.O. Box 5607 • Glendale, Arizona 85312-5607 • 1-602-978-2902 • Telex (via WUI): 6501267701

OFFICE: 2412 West Greenway • Suite B-10 • Phoenix, Arizona

July • August 1986

TIME DESIGNS MAGAZINE

FOR ALL TIMEX AND SINCLAIR COMPUTERS

TIME DESIGNS MAGAZINE CO.
29722 Hult Rd. • Colton, Oregon 97017
(503) 824-2658

Editor: Tim Woods

Assistant Editor: Stephanie Woods

Editorial Assistant/Production: D.L. Woods

Photography:
(unless otherwise noted): Thomas Judd
Printing by; Toad'l Litho Printing and Comp., Oregon City, Oregon 97045

SUBSCRIPTIONS: $15 a year for six issues (US funds only). No extra charge to Canadian subscribers. All other countries please write for information on air mail rates.

CUSTOMER SERVICE: Customer satisfaction is our goal. For subscription service problems please write or call TIME DESIGNS.

CHANGE OF ADDRESS: Write or call to prevent delay of sevice,

COMMENTS by the editor

Welcome to the "Serious" Side

A recent comment by Amstrad director, Alan Sugar, sort of "ruffled my feathers" a bit...and I'd be curious to get your reaction as well. To paraphrase, he stated that Sinclair computers were very games orientated, and that his own Amstrad models were for more "serious" applications. If you've been a Sinclair "addict" for a few years, as I have been, you might not have swallowed that line.

Have you also heard what they have planned for this coming fall? Amstrad has designed a games cartridge player based on Spectrum Z80 technology, but no keyboard or programming ability, and will be sold as a Sinclair product. I also understand that Activision (remember the cartridges for the Atari 2600, that ended up selling for $5 each in shopping malls?), will be jumping on the band wagon.

It's just sad to see our powerful little computers being panned as merely game playing devices, when there is so much more potential to be explored.

There does appear to be a striking difference between computing interests here and abroad. Games do make up about 90% of software sales in Great Britain, and if you've glanced through the top selling U.K. Sinclair magazines, you would understand where Mr. Sugar is coming from. About every other page has a full color advertisement for some new "mega hit".

Now don't get me wrong, I enjoy computer games very much. I just slapped down $48 hard-earned dollars for some more Spectrum games. The animation and graphics on some of them are extremely brilliant and a real treat to watch. Time Designs accepts and will print game listings of all types.

Over the past few years, our community of Sinclair users have struggled along with very little "outside" support. We have banded together, and have figured out ways of doing things that no one would have imagined possible with such "low-end" microcomputers. We have squeezed out every little bit of value that we could. There are now full-blown CP/M options available and with the appropriate interface, virtually any type of external peripheral can be added. Most of us continue to learn as much as we can from the examples that our fellow computerists have to offer.

Mr. Sugar...welcome to the "serious" side.

Tim Woods
"the editor"

see details on page 13

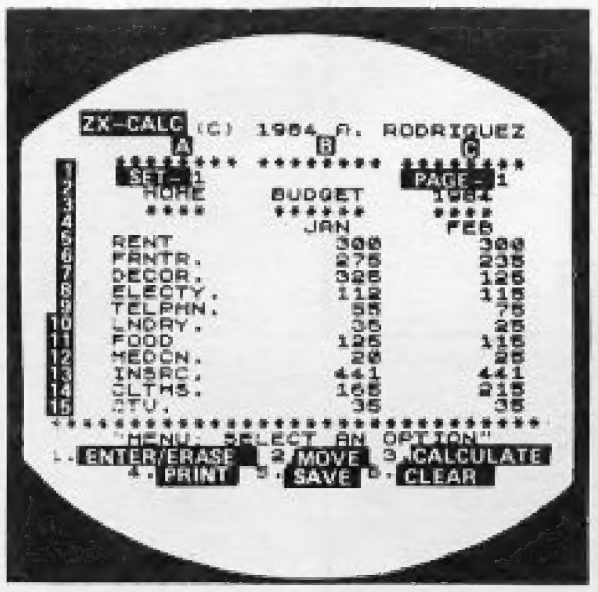
POWERFUL AND INEXPENSIVE BUSINESS SOFTWARE FOR ZX81, T/S1000 and T/S1500 COMPUTERS

ZX-TEXT

A word processor is to a computer user what a typewriter is to a typist, except that the former has more advantages than the latter. ZX-Text can operate in 16-64K RAM providing from 1300 to 6500 words per document. It features 6 different options: write, read, edit, print, save and clear text. Text is written on a per-line basis with quick speed and with horizontal back-space and delete capabilities being available. You can also access the editor directly from write mode and vice-versa. Text can be proof-read on a per-line basis allowing for enough time to determine if any editing is needed. The text editor allows a line of text to be deleted, inserted, replaced and listed for editing. You may also change a word or expression within a line, stop or start text while it is scrolling up the screen, begin reading text from the first line of the file, re-enter write mode from the editor, return to the main-menu or create a window so that you can read-edit two files simultaneously. The print option takes text displayed in 30-column format on the screen and outputs to either the ZX/TS printer. (With Memotech's Centronics Parallel Interface 80-column and lower/higher - case output is possible.) Files may be saved on tape cassette with the use of one single command, or by the same token they can be erased from memory / storage so that the full capacity of the program can be used for other purposes such as composing letters, reports, articles, memos, standard forms, instructions, ads, graphs, telephone directory, lists of customers, members, friends...etc. Also copies of files are always less expensive and easier to run than using a photocopier. Other advantages are savings in time, paper, ink, correcting mistakes and adding afterthoughts more efficiently than doing them through either handwriting or using a typewriter.

$16.95

ZX-CALC



An electronic spreadsheet calculator is the fundamental basic tool for summarising, reporting and analyzing in matrix form any accounting, mathematical or scientific manipulation of numbers. ZX-Calc operates in 32-64K RAM and affords a maximum of 3360 characters / spreadsheet. The entire matrix consists of 15 columns (letters A-O) and 30 rows (numbers 1-30) with 8 characters / cell. Unlike other popular ESCs, ZX-Calc uses in calculations and within cells all 14 math functions on the ZX-81/TS1000. It offers a unique *SUM function that totals one or more rows/columns simultaneously. Parenthesis can be used within equations. There is no fixed limit on how many equations may be entered. Formulas may be stored in all 420 cells of the spreadsheet. The display affords 15 rows/colums. Loading of data into more than one cell can occur across/down one or more row/column simultaneously. With vertical windowing you can arrange a set of columns in any order, or practice using fixed-variable-alignment display formats. The menu offers 6 options: enter/erase, move, calculate, print, save and clear the spreadsheet. Enter/erase allows the entering, deletion or data alignment within a cell through the use of a mobile cursor. With the move option you may move around the entire sreadsheet to access any row, column or cell. The calculate option allows you to enter labels, values or formulas into a cell or write and enter equations that will act upon the data already within the spreadsheet. You can also enter bar graphs into a cell in this option. Absolute/relative replication, down/across a column/row, is also allowed by this option. Also this option allows the automatic calculation of the entire spreadsheet with one single command. Print allows you to output to either the ZX/TS printer the entire spreadsheet by column-sets and row-pages through use of the COPY command. The entire spreadsheet may be saved on cassette tape or you may clear all data from it or erase the program from RAM entirely. The most salient advantage provided by an ESC over specifically vertical applications software is that an ESC provides a reusable framework with which you can compose any specific financial model rather than just be limited to only one statically fixed format for storing, displaying and manipulating numerical data.

$16.95

ZX-CALENDAR

Time management is an important aspect of any serious business and personal agenda. Planning how to spend our time leaves us better prepared before and while we are spending it and we remain better organized after we finish spending it. ZX-Calendar operates in 16-64K RAM affording 25 appointments in 16K, 100 in 32K or 180 in 48K and 64K. Each appointment record holds a maximum of 220 characters. The main menu includes enter, search/check/sort, change, save, clear and print any and all appointments made on a specific date or with any party. Output to either the ZX/TS printer is permissible. This program will permit you to remember to do something or to be somewhere important by cataloging your answers to six questions that you must account for in order not to waste time when it is scarce: when, with whom, at what time, for how long, where and what are you going to discuss and conclude when you get together with someone else? The program lets you permanently originate, record, classify, search, sort, calculate, modify, summarize, obtain a written report and store your answers to the preceding questions so that you will not forget what you decide to do with your time. This program identifies your time according to when you are going to spend it and with whom you are going to share it. Through these forms of labeling appointments you are able to verify or modify how your time is budgeted without wasting ink, paper or more time trying to remember what you said to yourself or what someone else said to you or where you placed certain written messages that you now can't find. With this program you will know where you can find exactly what you need to know about where you want to and have to be, or where you have been, before you get and after you got there. Thus, ZX-Calendar will let you plan your time so that you will never have to worry about what is ahead or what came before, for you will always know, by using it, to never be caught astray by any time-frame.

$16.95

$3.00 SHIPPING AND HANDLING/PROGRAM

A.F.R. SOFTWARE - 1605 Pennsylvania Avenue, No. 204 - Miami Beach, Florida 33139

DEALER INQUIRIES WELCOME **(305) 531-6464** FLORIDIANS ADD SALES TAX

L E T T E R S

"I have started a Users Group in the Harrisburg Area. I am having trouble locating users. The name of our Group is: Harrisburg Area Timex Sinclair Users Group. H.A.T.S. meets on the third Wednesday each month. I started the group with a few names I received from Russell Electronics. I originally visited the users group in State College, Pennsylvania, but that is too far to drive."

Sincerely,
Harrisburg Area TS Users Group
c/o Dave Bennett
329 Walton St. (Rear)
Lemoyne, PA 17043

Editor: Hope that printing your address here will bring in some more members for your group. I might add that we will print the address of any group that sends in the information.

To the editor and the readers,

"I want to introduce myself to you. My name is Charles E. Goyette. I live in the province of Quebec (Canada) where French is the spoken language. That is the reason why my way to write is so strange; I am not used to English.

I liked Time Designs so much that I decided to give my programs to the readers. CAVERN, the program you have seen in the last edition (May/June '86), was not, I admit, an excellent program. I had a few "bugs" and it was slow too. But in the month of May, I programmed the game SKI, which is listed elsewhere in this magazine. It is, I think, a pleasant game which has speed, and features good improvements compared to CAVERN. I hope you will enjoy it as much as I enjoyed programming it.

I must say on the other hand, that we have lost one brilliant programmer for the Timex computers...John Coffey [Editor's note: he is programming for the Atari ST line of computers now]. I imagine that it took him over 300 hours of work to create DIAMOND MIKE II, a game that all of us should have bought...there is always someone "crying" because there is no software for the Timex computers...here is a guy who is just doing that: software! But John, I wish you good luck in all your enterprises. And to you the readers, it's not too late to buy DIAMOND MIKE II. But don't wait..."

Charles E. Goyette
Drummondville, Quebec
Canada

Editor: Charles, I really enjoyed playing SKI the other night--it's addictive and really gave the flavor of down hill (Alpine) skiing (a favorite past-time during the winter months here in the northwest). One of our readers requested an explanation on how to program the little "spaceship" figure in the CAVERN listing (and probably the figures in SKI as well). Maybe you could enlighten us with a programming tip for the next issue?

I had a chance to speak briefly with John Coffey at the T/S Computerfest in Cincinnati. Although he praised the Atari ST's, he mentioned how easy it was to get good results from his Timex/Sinclair. I have a "hunch" that we might be hearing from him again. Also note that Diamond Mike II is available from JRC Software, WMJ Data Systems, RMG Enterprises, and other dealers as well.

"I read with some interest, in the March/April '86 issue of Time Designs, The problems James F. Brezina had with the fact that the function INT (SQ(675)) equaled 25. He felt there was something wrong with his 2068, and your reply stated that your 2068 did the same thing, and the "flaw" was in the ROM. In actuality there is no flaw. All computers use some sort of "successive approximation" to evaluate functions such as the square root. It carries the approximation to the point that successive answers are, to the accuracy of the computer, equal. If you will ask the 2068 to PRINT 26-SQ(676), you will find the answer is something like 7.45E-9. Since the computer is accurate to 8 figures, this small residual is treated as zero. That is, it is treated as zero by the computer but not by the INT function which truncates to the next lower whole number.

This is not a problem unique to the 2068. I have worked on very large computers, and had essentially the same problem. The lesson to be learned is that branch points should be tested before running a program. There are a number of ways to circumvent the problem. Mr. Brezina used one in changing the store number to 687. A second is to add .5 to the number for which the integer value is desired. This assures the number is rounded, and not truncated. A third method is to provide a range for the branch value. For example, if the branch test is for some function to equal 0, then the instruction should be for the absolute value of the function to be less than some small number such as .0001.

I hope this information will aid others who are writing programs to avoid a pitfall that took me over a week to discover the first time I ran across it. And reassure 2068 users that Mr. Brezina's problem does not indicate a flaw in their machine.

Very truly yours,
Vance J. Carpenter
Fairport, NY

Editor: Perhaps "flaw" was a poor choice of words... thanks for setting the record straight, and sharing your tips on correcting 2068 mathematical routines--a subject that is briefly discussed in the users manual.

"As an additional bit of trivia on Richard Hurd's keyboard article, try a coating of clear fingernail polish to the keys. This will stop the wear-out of the letters. I have tried several brands and have found that Max Factor #13 Clear is the best. It leaves a clear transparent coating. I have two 2068's...the first I wore the letters off the keys in nothing flat. The second one I coated, and it is still going strong after a years hard use. E. Arthur Brown has key legends with sticky backing for $1.50 a set. I got one and they seem to be real nice."

P.S. Amen to Dunnington's "All Caps Please", I am an old duffer with trifocals.

Dudly S. Rea
Priest River, Idaho

Editor: Mr. Rea also enclosed a circuit diagram for adding automatic two-key entry from a one-key closure, to answer a question posed in Richard Hurd's external keyboard article (see the May/June issue). Ran out of space, but we'll print it in the next issue.

Sinclair Micro Update

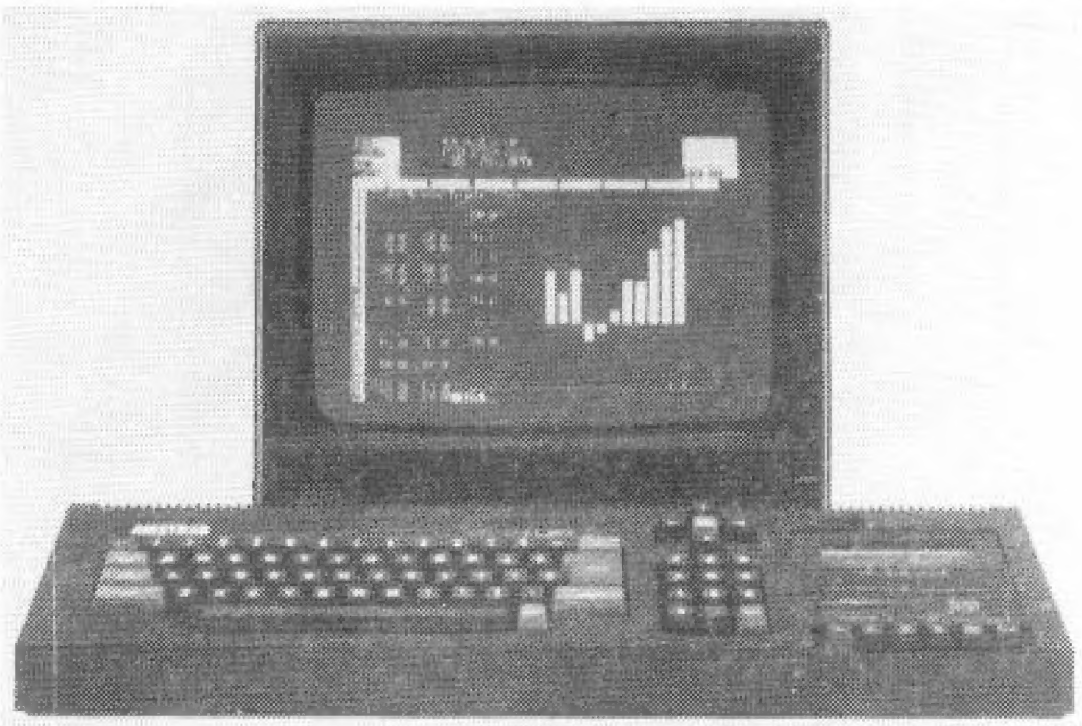
Sir Clive's Spectrum 128 will now go the way the "rubber-keyed" Spectrum's went..."out the door" and priced to sell. The Spectrum 128 must make room for the new SPECTRUM PLUS TWO, which Amstrad has already put into production for the upcoming Christmas season. The new Spectrum will be a repackaged 128, with two genuine joystick ports and a cassette tape recorder built-in.

Industry watchers have commented on the treatment that Amstrad is giving the Sinclair line. One particular expert remarked, "The Spectrum is a full fledged Amstrad now..." It is true, that the similarities between the new Spectrum Plus Two and the Amstrad CPC 464 are remarkable.

WH Smith's and Dixons (large retail chains) have placed the Spectrum 128 on sale at around £139, and usually packaged with game software or an inexpensive dot matrix printer. The Spectrum Plus goes for about £109 with "extras", and the British QL is priced at £200 and includes a printer.

In further developments, Sir Clive Sinclair will go ahead and develop the long-awaited portable PANDORA on his own, with a yet un-named financer. Amstrad has expressed no interest in the computer, as it would most likely compete with existing Amstrad models. The Pandora has been announced as using Flat Screen Television technology, but Sir Clive has had a change of mind lately, and is looking at liquid-crystal monitors.

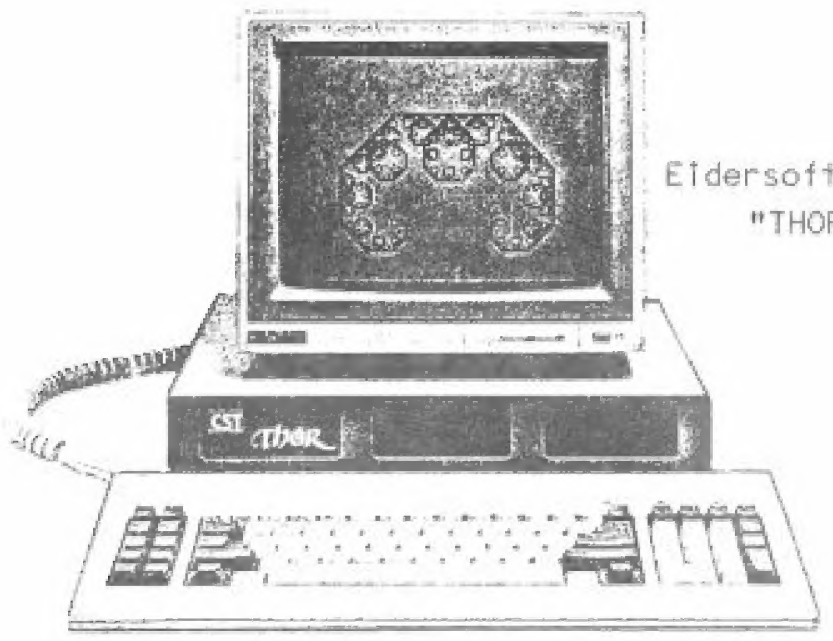
Just before the Sinclair "break-up", another computer had been on the drawing board, code-named LOKI, and nick-named the "Super Spectrum". It was rumored to use custom display chips in order to function much like Commodore's expensive Amiga, and have 256K on board RAM. It is not known at this time, if Amstrad will use the Loki technology, of if Sir Clive will market it himself.



Above: The Amstrad CPC 464, step father of the Spectrum Plus Two. Right: Sir Clive Sinclair--new micros in the works?

Meet the QL Clones
Support for QL Continues

The QL will continue on as a professional small-systems computer, due to three possible "clone" units slated for production soon. The first to be announced is the CST (a disc manufacturer) and Eidersoft (software company, best known for "ICE") QL based machine called "THOR". It includes two units. One housing the QDOS, 3.5 inch floppy drives, 640K RAM, all Input/Output ports and built-in ICE (of course), while the other unit will be a separate IBM-style keyboard attached to the main Thor unit via a telephone cable. Many options are to be planned such as a 68020 processor board, and a 20 mega byte hard disk. The basic model will start at around £550.



Eidersoft/CST's "THOR"

Another QL-based microcomputer is being developed by QDOS author Tony Tebby, and called the "QLT". He will be developing and manufacturing the machine with another company, who will have equal shares in the investment.

A third QL is rumored to be in the works by an un-named American company, who will have the computers manufactured by original QL factories, Thorn in the U.K. and Samsung in Taiwan. The manufactuing rights were being negotiated between the American investors and Amstrad as we went to press.

In a recent gathering of QL software and hardware vendors and manufactures, a general concensus was expressed to continue supporting the QL indefinitely. The caucus met in London, England, and included some representatives from the U.S. Several software houses have now "scaled-down" their operations, and are now based on mail-order sales.

Super ZX81 Support from Scotland

ZX-81 and T/S1000 Users--even though Software Farm (the short-lived ZX software company and support group) of Great Britain has "bought the farm" (as a manner of speaking), there is an excellent source of information and support that you might not be aware of (and just when you thought that the computer world had ignored you for good).

Nick Godwin of Berwickshire, Scotland, has programmed a very extensive "ZX81 EXTENDED BASIC", that is reported to rival the Spectrum. ZXEB is supplied on tape complete with a ten page users manual. The Extended Basic utility contains 44 additional instructions, including special graphics commands such as FLASH, and machine code routines such as CALL, which permits any user-specified code routine to be called directly from Basic (whether in ROM or RAM). The price of ZXEB is £10.

Also available is an outstanding bi-monthly newsletter called "ZX Broadsheet". It is simply packed with programs and programming tips. ZX Broadsheet has been around for about five years now, with the latest issue running ten pages. A sample issue is priced at £1.50.

When corresponding with Nick Godwin, be sure to include a S.A.S.E. (with the proper over-seas return postage). Send inquiries to: Nick Godwin, 4 Hurkur Crescent, Eyemouth, Berkwickshire, Scotland, TD14 5AP.

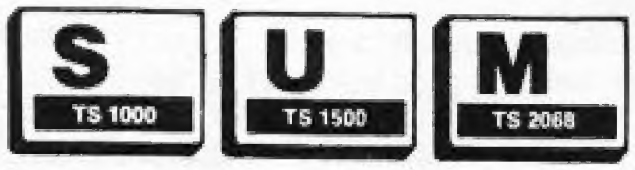
TIME DESIGNS Acquires SUM

Time Designs Magazine Company has acquired SUM Magazine of Gainesville, Florida. SUM (The Sinclair/Timex Users Magazine) has been published for two years, but was previously the newsletter of the T/S Users Group in Gainesville (founded in September 1982). The talents of Joe Williamson, the editor, and Richard Cravy, the publisher, produced a very attractive Sinclair magazine/newsletter, that was mailed on pretty much a monthly basis, and for the most part, was published on time (an unusual occurance in our particular industry).

Time Designs has decided to absorb SUM into TDM, instead of publishing two separate magazines. As a result, Time Designs Magazine will be physically larger, and will have the largest circulation of any related magazine in the United States. All previous subscribers to SUM will receive Time Designs, to fulfill their present subscriptions. This will be conducted on a bi-monthly basis as opposed to the monthly format of SUM. All subscribers who received both magazines will have their subscription to Time Designs extended accordingly.

We welcome all SUM subscribers, and would like to mention that we are very committed to publishing a quality magazine for Sinclair enthusiasts. All comments, questions, or subscription service problems can be forwarded to us at--TDM, 29722 Hult Rd., Colton, OR 97017. We would also like to wish Joe Williamson and Richard Cravy the best of luck! Although SUM readers can be assured that they have promised to contribute an article now and then. Be sure to read the following letter from Joe.

--Tim Woods



LETTER FROM THE EDITOR OF SUM MAGAZINE

We at SUM Magazine would like to thank all of you for the support you have shown over the past years. The activity in the Sinclair/Timex market seems to be ever growing, and it is exciting to see so many of you sticking with what many people considered to be a "dead" computer. This tells you something about the "power" of our computers that others just don't see!

Richard and I have found that you can't take the work that goes into this magazine lightly. What used to be just a simple part-time job has grown into one that requires full time attention. In the meantime, both Richard and I have taken on more and more responsibility in our full-time jobs and are finding it more and more difficult to find the time it takes to bring you the quality magazine that you have come to expect.

With this in mind, we have worked out an agreement with Time Designs Magazine, where we will merge with them to bring you the largest and most comprehensive magazine in the U.S. Sinclair/Timex market. We, along with the staff of TDM, will continue to bring you the quality reading you have come to expect. The July issue will be the last issue put out by SUM Magazine.

Time Designs Magazine will cover all subscriptions on through to their expiration date. Those who have a subscription to both SUM and TDM will have their subscription extended by the amount of time they have left with SUM. So you will continue to have uninterrupted service of the best resource for your computer around.

Time Designs Magazine is dedicated to the Sinclair/Timex market with coverage on all Sinclair/Timex computers and they are continually expanding. They have top notch writers like Wes Brzozowski, Bob Orrfelt, and Michael Carver and publishing is their only business.

Tim Woods, the editor, along with his wife, Stephanie, the assistant editor, are looking forward to bringing you the best in news, articles, programs and projects available.

Richard and I are sorry that we had to come to this decision, but we both feel that it is the best for everyone involved and we hope that you will understand and support our endeavor to serve you best.

Thank you,
Joe Williamson
Richard Cravy

P.S. Congratulations to Tim and Stephanie on the birth of their son, Timothy Dean Woods, born on June 10th at a healthy eight pounds!

Product/Dealer News

Gulf Micro Electronics, 1317 Stratford Ave., Panama City, FL 32404, has available a comprehensive software package on either cassette or special expanded version on disc for Aerco FD-68 users. Entitled SMART TEXT TS-2068, the author, Bill Jones, refers to the package as "Administrative Software". There are four operating programs, including a Data Base, a Word Processor, a Mailing List Manager, and a special Printer Patch program. Disc version comes with an automatic, self-adapting version of Printer Patch, and a Program Tutor file. Both versions come with full documentation. Price $34.00 ppd. When requesting information, ask about new versions for the Oliger Disk System and Zebra's OS-64 Cartridge.

Speaking of Aerco's popular disc system, there is a specialty user group catering to this system and a newsletter which is publsihed quarterly. Cost for a one year subscription is $15. For information, write to: David Hill, 1159 S. Shore Dr. #12, Holland, MI 49423.

You might also consider subscribing to a cassette-based magazine for the T/S 2068 called BYTE POWER. Each tape has programs ranging from Arcade games to Business programs. There are also reviews and programming tips. One tape (sample issue) is $5.50. Six issues, $29.99, and 12 issues for $49.99. Send check or money order to: Byte Power, 1748 Meadowview Ave., Pickering, Ontario, Canada L1V 3G8.

Sprite graphics, the key to successful game programming is an area that hasn't been addressed too often for the 2068. Now two programmers (from separate states) have collaborated on a new software development package called SPRITES 2068. It contains several machine code utilities, demos, and a 34 page manual. Priced now at $19 ppd. Send check or money order and inquiries to either: Vern Tidwell, 1303 Whitehead St., Key West, FL 33040, or Ron Ruegg, 37529 Perkins Road, Prairieville, LA 70769.

Beaver Computer Products, 999 Munroe Ave, Winnipeg, Manitoba, Canada R2K 1J4, the company that features "extended video mode" software for the T/S 2068, has some new titles. "Beaver Writer" is touted as the first 80 column word processor for the 2068, and "Character Font Generator" lets you add character (pun intended) to programs and text. Prices: Beaver Writer, $25 (U.S.), Character Font Generator, $15 (U.S.). A catalog which includes a demo tape is available for $1.50 (U.S.).

Some very exciting software has been developed by another Canadian company called Novelsoft (106 Seventh Street, Toronto, Ontario, Canada M8V 3B4). Some of you may be familiar with David C. Ridge, who has had his ARTWORX marketed in Great Britain for the Spectrum, and is currently the Senior Programmer for Novelsoft. Now there is an improved version of his popular graphics package for the T/S 2068 called ARTWORX version 1.1. It is priced at $19.95 (U.S.) + $3 postage. Another program being offered for the 2068, and should sell quite well here in the states, is a Basic Compiler called TIMACHINE and is reported to outperform any compiler on the market today for the Timex. Timachine will handle all Basic commands (except I/O), and will convert your program to fast machine code in seconds. The program is priced at $19.95 (U.S.) + $3 postage.

A.F.R. Software, 1605 Pennsylvania Ave. #204, Miami Beach, FL 33139, has three software programs for the T/S 1000/1500/ZX81 (and versions for the 2068). ZX-TEXT is a word processor, ZX-CALC is professional spreadsheet program and accounting model package, and ZX-CALENDER is time-management program. All three titles are priced at $16.95 each + $3 postage.

BF Kimbrough KEL "In-Memory Operating System Ver. 1.0" for the T/S 1000 and ZX81, is an interesting software utility. It is written in relocatable machine code and operates in BASIC or user defined area. The operating system also features ten user-definable function keys. Price: $7.97. Send check or money order to: BF Kimbrough, 723 Roselle Ave. Flr 2, Akron, OH 44307.

COMLINK I is an RS-232 serial communications interface for the T/S 1000 and ZX81. All software is in EPROM for instant loading, and COMLINK I can be used with any 300 baud modem. All operating power is derived from the Sinclair. The advanced software is menu-driven and has many features including user defined Macro keys, auto-repeat, expanded character set, and more. For further information and prices, write to: A. Eckhardt, 918 Anna Street, Boalsburg, PA 16827.

Curry Computer, PO Box 5607, Glendale, AZ 85312, has obtained the exclusive marketing rights to an outstanding line of software developed in France. Pyramide Software for the QL, is popular in Europe, and has now come to America (thanks to Curry). WANDERER is a 3-D space arcade game that requires the user to wear the supplied red/blue glasses. VROOM is a racing simulation. The driver sits in a Grand Prix racer, and maneuvers around five different tracks. QL-PEINTRE is a graphic-design package that is very similar to MacDraw and MacPaint. OTHELLO is a 3-D (no glasses required with this one) version of the classic game. Write to Curry for a complete catalog with prices.

PCIMPORT is a program that permits your QL to download ASCII files from an IBM PC via direct link. This permits the transfer of documents, program source code or any other ASCII encoded file from the IBM PC to the QL. Also included is a conversion program that converts Micro Soft Basic to QL Super Basic. For a catalog of QL items and prices (including PCIMPORT), write to: MIN-NY Electronics Inc., 7332 Douglas Dr., No. Brooklyn Park, MN 55443.

A+ Computer Response of Keene, New Hampshire, has added five new American QL dealers to their list, making a total of 17. The new dealers are: Markel Enterprises, PO Box 2392, Secaucus, NJ 07094; C.W. Associates, 419 N. Johnson St., Ada, OH 45810; Variety Sales, 325 W. Jersey St., Elizabeth, NJ 07202; Quantum Computing, 8 Gillen Street, Mine Hill, NJ 07801; and Info-Mation, RR#1 Box 260, California, MO 65018.

The Second Annual Mid-West Timex/Sinclair Computerfest will be held in Indianapolis, Indiana on the first weekend of May 1987. The core of organizers for the Cincinnati show are currently slating plans for the 1987 "reunion" of dealers, exhibitors, and Sinclair fans from the mid-west and virtually everywhere else. If you would like to obtain some preliminary information...write to: Frank Davis, 513 East Main St., Peru, IN 46970 (send a S.A.S.E.) or call (evenings) 317-473-4885. There has been interest expressed in T/S Computerfests for the New York/New Jersey area, and for the west coast (possibly San Francisco?), but so far, nothing definite has been planned.

"Commrades...all together now...enter the keyword [PRINT] and followed by CARL MARX in quotations." And its all for the good of the party! Whats going on here? The Polish government is about to receive 800,000 Timex 2068's and 200,000 FDD-3000 Dual 3" Disk Drive Systems, to be used in public schools and institutions. The "iron curtain" deal was recently struck between the Timex Corporation and a Polish industrial firm (through a neutral distribution agency. O.K., now how many issues of TDM should be shipped?

The temporary shortage of three inch (Hitachi type) floppy disks is over...and supplys are very good. The following companies have the "special" disks in stock for immediate shipping: Zebra Systems Inc., (718) 296-2385; Peripherals Direct, (312) 498-9244; Speedysoft (London, England) 01-789-8546; various other dealers around Great Britain. Resulting from a recent deal struck between Amstrad International and Sears, various selected Sears outlets will carry the 3" disks.

MARKEL BELIEVES IN THE QL

Mark L. Fendrick is happy to announce that MARKEL ENTERPRISES is now an authorized QL dealer. To celebrate, TIME DESIGNS MAGAZINE subscribers can take $25.00 off the purchase of a 128K QL computer. (These are true American models and carry the U.S. warranty.) All QL's come with QL Quill (wordprocessor), QL Archive (database), QL Easel (Business Graphics), and QL Abacus (spreadsheet). The QL has two built in drives and can be used immediately after unpacking.

To get your discount send the coupon below with your subscription label

QL 128K computer with software.....$299.95
QL RGB color monitor (cable included).....$299.95
QL printer (serial with Near Letter Quality print).....$299.95

Combination I...QL w/either monitor or printer.....$575.00
Combination II..QL w/printer AND monitor.....$795.00

SPECIAL DEALS FOR COMPUTER SCIENCE STUDENTS. WRITE OR CALL FOR DETAILS.

Add 3% for shipping and handling. Street address needed for shipping.
NY and NJ residents add appropriate sales tax.
Sorry---no credit card orders---yet.
More items available....write for catalogue.

---MARKEL ENTERPRISES---

$25.00 off — $25.00 off

With the purchase of any QL or QL combo

TIME DESIGNS MAGAZINE SUBSCRIPTION LABEL REQUIRED

- One coupon per customer -

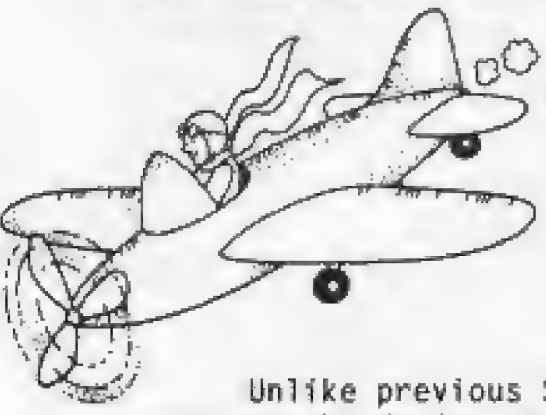
expires August 31, 1986

$25.00 off — $25.00 off

---MARKEL ENTERPRISES---

MARKEL ENTERPRISES/P.O. Box 2392/Secaucus, NJ 07094-0992/(718)627-1293

Games for the Sinclair QL: A Potpourri



by Mike de Sosa

Unlike previous Sinclair computers, the Sinclair QL was not--by design--optimized for computer games. The sound and graphics systems of the QL are adapted for professional and business use, partly in an attempt by Sinclair Research to live down a reputation as a producer of computers for children. But the innate flexibility of the QL's CPU, QDOS, and SuperBASIC permits certain types of computer games to be programmed which are second-to-none. Psion's QL-Chess, the world's microcomputer chess champion in 1984, is, in every way--playing strength, flexibility, appearance, and user-friendliness--absolutely superior and a very good buy. It can provide chess fans at any level, countless hours of delight, instruction, and analysis (of chess problems, tournament games, ect.). Psion's Match Point is another example of their formidable prowess in programming life-like computer simulations--in this case, tennis.

QL-Chess is a comprehensive, realistic simulation of the ancient game featuring 3-D pieces and playing board (no glasses required), 28 levels of play from novice to champion, chess clocks, too many analysis functions to describe in a short article, and numerous playing options. Get it. You'll love it. One of the best computer programs of all time, a steal at $25. (At today's prices, the combination of the QL and QL-Chess alone--never mind the QL's other uses--make it an exceedingly good buy; it is ironic, perhaps, to consider how well-received such a combination might be in a chess-playing nation such as Russia.)

If you use a QL with additional RAM and QL-Chess will not load correctly, try deleting lines 110 and 120 of the BOOT program on your backup copy of the program cartridge. This shouldn't work, but it does for me. Some programs don't work with a full 640K RAM on the QL, so, if like me you hesitate to remove a difficult-to-install RAM card before loading a game, inquire of your dealer whether a given game will work with your QL set-up.

Psion's Match Point is a tennis simulation featuring extraordinary graphics (even the shadow of the ball on the playing surface is true-to-life, as are the ball boys and girls who retrieve net balls). A game for one or two players, Match Point provides the opportunity for both strategy and dexterity--you can decide whether to employ a baseline or net game on each service and, depending upon the movement of your player and the timing of his swing, impart various kinds of "stuff" to the ball.

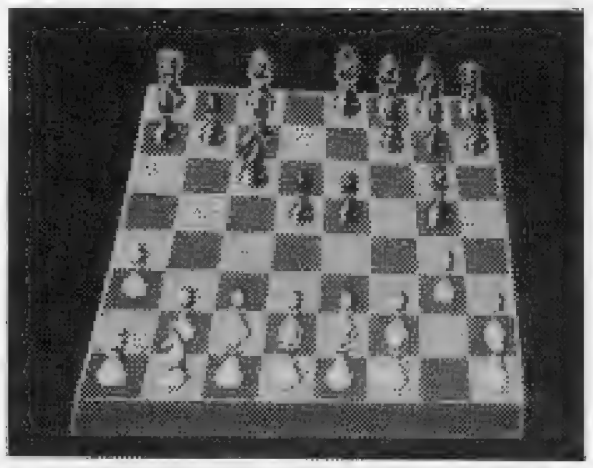
As is the case with QL-Chess, the rules of the computer game are the same as those for the real game, and play is only slightly less difficult. Unless they are really virtuosi, small children may find considerable difficulty in returning balls. A stick is a necessity--play with the cursor and SPACE keys is a near impossibility. (The Archer Deluxe Competition Joystick, available from Radio Shack, is a good economical choice for this and other QL games.)

Match Point is visually attractive and challenging (demanding might be a better word) enough to hold an adult's interest over time. And, if you find that play--even at the quarter finals level--is too tough for you, it is always a pleasure to watch the computer a computer demo game at the finals level.

After the Psion games, there is, with few exceptions, a distinct drop-off and a wide variance in the quality of "lesser" games. In what follows, I will only discuss the relatively few QL games that I've actually tested, so, if I have left out your favorite game, forgive me and consider touting it yourself, in a letter to the editor.

War In The East by Mark Steuber of SHARP 's INC., in Virginia, is the first major American game for the QL and is, in its way the equal of Psion's games. A military strategy and tactics game like its Avalon Hill and Strategy and Tactics forebearers, but with all of the time-consuming and laborious tasks fully automated--you don't even need a pencil--War In The East is a simulation of the Russo-German conflict during World War Two. Three scenarios are available: Scenario I, "Barbarossa 1941", dealing with the epic Nazi invasion and near conquest of Soviet Russia before and just after Pearl Harbor; Scenario II, "Stalingrad 1942", dealing with the second summer of the Nazi-Soviet conflict and the psychological turning point of World War Two; and Scenario III, "Destruction of Army Group Center 1944", dealing with the German Army's Gotterdammerung attempts to withstand overwhelming Soviet superiority at the end of the war. War In The East comes with Scenario I included; the other two scenarios are available for purchase separately.

Taking six or seven hours to play, War In The East may not be a game for everyone, but it represents an extremely accurate representation of the several campaigns--so much so that it should be very useful in teaching mobile land tactics and strategy to military officers in staff colleges. I don't know of a comparable personal computer game in this regard. At move 14 when



screen display of Psion's QL-CHESS

the rainy seasons begins, you can almost feel the tug on your boots as General Mud takes over, and at move 17 when the screen background turns white, you can almost feel the ground harden and the icy blasts as General Winter takes field. For a better appreciation of the game, I recommend that interested players take the time to review the historical record of the campaigns simulated in the various scenarios. There is only one problem--one which Mark Steuber should soon correct with a patch to the BOOT program: the game will not load on a QL with a full 640K of RAM. Highly recommended.

WANDERER, by Pyramide of Paris, is a state-of-the-art, 3-D, space adventure game--you actually wear a pair of cardboard spectacles with one red and one blue lens--in which you maneuver your space craft, attempt to evade or destroy hostile spacecraft, test your wits in a poker like exercise, explore black holes, ect., all in a supreme effort to rescue your neighbor's cat!

Subtitled, "The Planets Play Poker", Wanderer is unique in its 3-D line graphics and is probably good enough to hold your interest, even without glasses. (If you wear glasses yourself--or even if you don't--consider using clear colored plastic, a product available at large stationers' called Chart-Pak, or even the plastic in the cardboard specs that come with the game taped to your glasses' frame.)

Wanderer's documentation is inadequate, so here are a few tips. You replace your spacecraft's fuel and win game points by zapping other space vehicles--friendly, neutral, or otherwise, I believe, but you can convert a neutral to an enemy by firing at it. TIP: save your "cats" (the unit of currency), for extra or replacement shields which protect your craft from destruction while they last; do not plan to spend your cats for "energy". There is no telling who your enemies are until they fire, but there is an alarm to warn if you're currently being shot at and missed. Your spacecraft is equipped with two side-by-side radar scopes: the left radar portrays left-right and up-down indications of other spacecraft (your spacecraft is at the center of both scopes); the right radar depicts left-right and ahead-behind data. Final aiming is "from the hip". There is no gunsight for precision aiming.

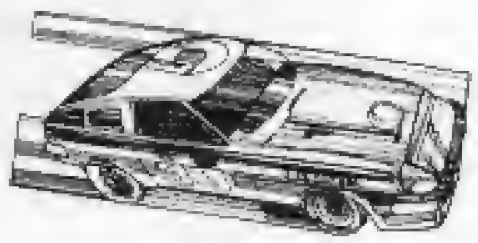
How does Wanderer stack up? Graphics, impressive; space maneuvering and firing, good; documentation, not so good--there is a language problem; game concept, fair; prolonged ability to hold your interest, probably not great, but then Wanderer is not my kind of game.

Squadrons is my kind of game, a simulation of one day's air combat during the early phase of the Battle of Britain in 1940 by Peakcrown Ltd. In this simulation, you are at an RAF air defense control center with the responsibility to launch and vector fighters from six air fields against multiple German air raids launched across the British Channel and North Sea against Southeast and South Central England, including London. Some German raids are feints or deceptively routed in concert with other raids, some are designed to attrite "the few" RAF pilots, knock out British radars or airfields (which limits your ability to defend), still others to attack population and industrial centers. Your specific duties as RAF Group Air Controller are extensive: you must decide when to launch fighters, and how many to launch; decide which German "raids" to attack; steer fighters in against these raids from the proper direction (up sun) and at the proper altitute (not too low or too high); break off attacks and recover fighters when necessary and in order to recover them safely before they run out of gas--no mean task during the "fog and friction" of combat (RAF fighter "combat" losses are a function of relative numbers of aircraft, height and directional factors, and ammunition expenditure); establish air patrols in case of radar outages; direct anti-aircraft units defending fighter airfields to "standby" and "stand down"--you can, if negligent, lose RAF fighters

to "friendly fire"; monitor airfield status and deal with such things as communications outages, runway damage, and clearing the field for further operations; monitor a myriad of incoming messages and take the appropriate actions; and notify all cities in the area of air raid warnings and "all clear" signals--yes, you are graded in this latter respect by the number of civilian casualties incurred and factory production hours unnecessarily lost. And if all of this were not enough, something like spinning plates on television, time is compressed: 12 minutes equals one hour.

Squadrons offers five levels of play in a game lasting as long as three hours (a "pause" function is available for when the phone rings). Considerable latitude is offered in the way of tactics and strategy. Should RAF fighters attack heavily defended formations? Should you, as Group Controller, make your first priority the defense of your cities or the preservation of your defensive force? At what stage of their sortie should fighters be recovered to minimize losses due to communications outages or running out of gas? Should air patrols be employed in certain key sectors, ect., ect.? Squadrons' graphics are perfectly adequate but not spectacular. Highly recommended. Not for hypertensives.

QL Hyperdrive, by David Woodward and The English Software Company, is a one-player, 100% machine-code grand prix motor racing game. Its features include 3-D perspective graphics, five skill levels, music and sound effects, and road hazards in the form of oil slicks. To proceed to the next higher stage (skill level and roadway), you must pass 20 cars within a short time limit. As a grand prix driver you may throttle, brake, and steer right or left.



In QL Hyperdrive's advanced stages, other cars swerve toward you necessitating prompt evasive action. I recommend use of a joystick, but play is feasible and not too bad using the cursor keys. It takes some practice and an average speed of about 90 to qualify in the first stage. As it is, smaller children may find it difficult to qualify for higher stages, but should still enjoy steering right or left and accelerating and breaking as they weave between cars at slower speeds. QL Hyperdrive does not load on my 640K RAM QL, and, although it should be a relatively simple matter to overcome this, I was not able to do so in the time available. Recommended.

QL Bridge Player II is a one-player contract bridge simulation by CP Software. It employs the somewhat quirky British ACOL bidding system which many American players may find annoying. Computer play is at times weak and inconsistent, and there is no analysis of bidding or play, that is, it does not explain reasons for a given bid or play. But all of these shortcomings are easily corrected. If our QL distributors squawk loudly enough, CP Software may replace QL Bridge Player-II with the much improved Bridge Player III, now available for the Spectrum. As it stands, QL Bridge Player-II, though somewhat entertaining, is not recommended in its present form.

1986 Thomas B. Woods Award. In the March/April '86 issue of Time Designs, I offered a $10 cash award to the first reader to come up with a program for the QL comparable to the following old Sinclair program and using ten or less program statements:

```
10 INPUT "Formula? "; n$

20 PRINT n$;" = "; VAL n$

30 GO TO 10
```

Sinclair SuperBASIC does not include the VAL function.

There were no submissions, but, for those of you interested, I did figure out a QL procedure that would produce the result. Since it entails using a Microdrive, disk, or RAMdisk file and the input/output of information, it works best with RAMdisks. The following program listing uses a Microdrive. To use RAMdisk, change the indicated program lines to read as follows:

```
5 FORMat ram8_10: OPEN #3,ser1

9 OPEN_NEW #4,ram8_work

12 MERGE ram8_work

13 DELETE ram8_work
```

Figure 1.

```
1 DEFine PROCedure VAL
2 CLEAR: CLS
3 INPUT "Hard Copy? (Y/N): ";p$
4 IF NOT p$=="y": GO TO 6
5 OPEN #3,ser1
6 PRINT "Enter math expression or 'z' to exi
t VAL"
7 INPUT f$;
8 IF f$=="z": CLOSE #3: END DEFine
9 OPEN_NEW #4, mdv2_work
10 PRINT #4,"14 y = ";f$
11 CLOSE #4
12 MERGE mdv2_work
13 DELETE mdv2_work
14 REMark
15 PRINT " = ";y
16 IF p$=="y": PRINT #3,f$;" = ";y
17 GO TO 7
18 END DEFine
```

Include the procedure in your BOOT or other programs or in a collection of defined function and procedures. Call the procedure by keying and entering VAL, then follow screen instructions. The VAL procedure will evaluate all mathematical expressions possible using SuperBASIC, including those with root, power, and trig functions and those using multiple sets of brackets. The procedure gives you a printing calculator result on the screen or your printer. I think you will come to find it one of your most useful SuperBASIC procedures.

Next time: A surprise! I have requested the use of some hardware for the QL and will discuss whatever turns up of general interest.

WANDERER, SQUADRONS, QL HYPERDRIVE, and QL BRIDGE PLAYER II were obtained from Curry Computer for the QL GAMES article. QL-CHESS and MATCH POINT were obtained from RMG Enterprises. WAR IN THE EAST was obtained from Sharp's, Inc.

curry computer

"We know the QL ... We've been appreciating it longer."

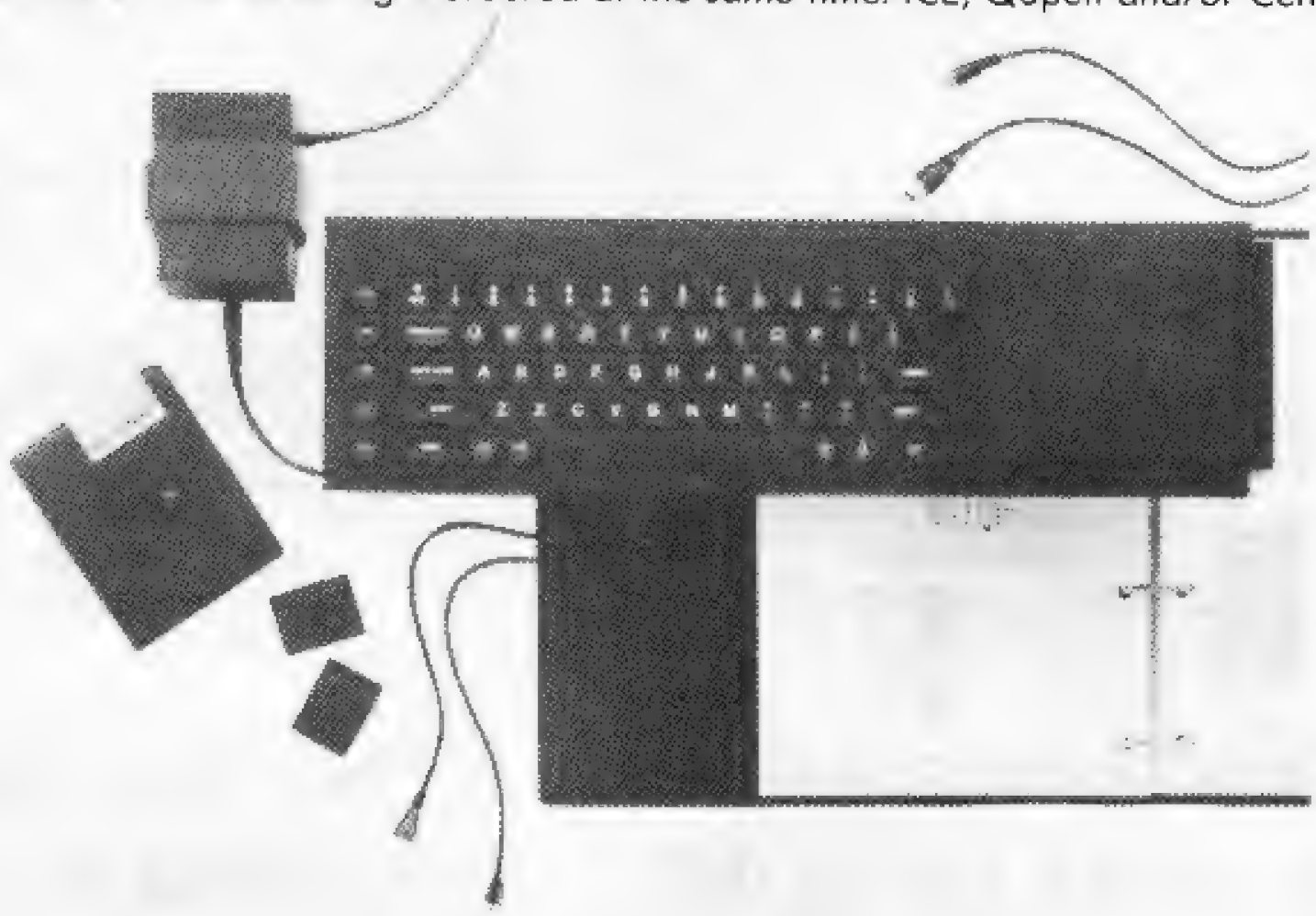
QL Special #1 — $289.95

>QL Computer-free ground shipping
>WD Utility Prg. worth $16.95
>Subscription QL Report $15.00
>8 Free programs including:
— music program
— windows demo
— graphics pictures
— calculator prg.
— and more

QL Special #2 — $299.95

> QL Computer
> Free I.C.E. ROM
I CON-based front end operating system for QL
> $8.00 shipping
Both packages include the 4 Psion programs.
These same 4 prgs. are also avail. on the IBM and retail for $695.00.

— Take 20% off the following if ordered at the same time: ICE, QSpell and/or Centronics I/F



While other computer companies talk about bringing the power of computing within reach of ordinary people someday soon, the Sinclair QL puts extraordinary power into their grasp, today. The power of the 32-bit architecture of the Motorola 68008 microprocessor. Of 128K RAM, expandable to 640K. Of two built-in Sinclair Microdrives for mass storage. Of networking. Of a full-size QWERTY keyboard. And of an operating system that accommodates windows and even multi-tasking.

The QL delivers up to 512 x 256-pixel full-color resolution or black-and-white display. It can be used with either a monitor or an ordinary TV. It has two RS-232C serial interfaces and accepts joystick cursor control.

Included with the QL are four sophisticated, powerful and easy-to-use programs. A word processing program with the power and simplicity of dedicated word processing, that lets you print exactly what you see on the screen. A database management program with up to 256 data fields, in which you can define relationships, extract data and reorganize it, do statistical and accounting calculations, and more A spreadsheet program so simple to use you can do a complete "what-if" calculation in a single keystroke. And a graphics program with 32-bit speed to create spectacular and immediate effects for home or business.

The price for all this power? That may be the QL's most powerful feature of all.

P.O. Box 5607 • Glendale, Arizona 85312-5607 • 1-602-978-2902 • Telex (via WUI): 6501267701

OFFICE: 2412 West Greenway • Suite B-10 • Phoenix, Arizona

QLiqlak

COSMOS QL Software Review by Paul Bingham

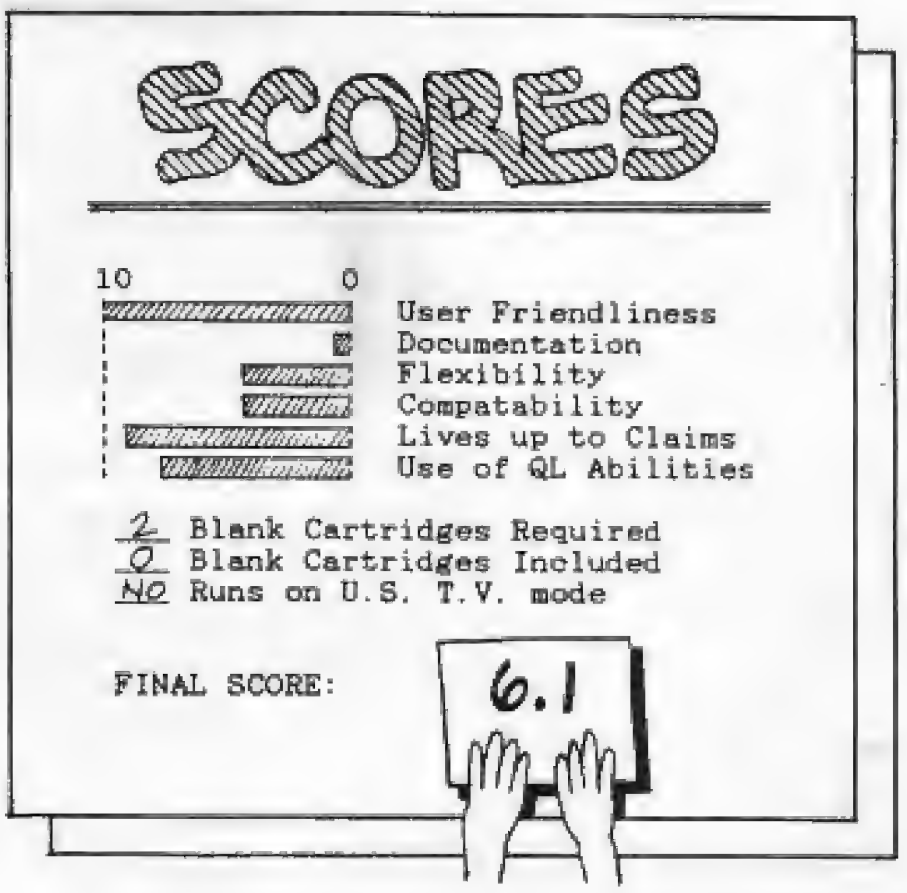
It is my hope that 2068 owners will read this review as well as those with QLs. A review of ASTRONOMER for the 48K Spectrum appeared in the January/February 86 issue of TIME DESIGNS, and in it I extolled its virtues. 2068 owners will be pleased to hear that ASTRONOMER II is out and includes many facilities not included in the original. I will necessarily be doing some comparisons between ASTRONOMER and COSMOS for the QL.

COSMOS is written by G.F. Cornwell for Talent Software in Scotland. I must say that COSMOS takes longer than any other QL program I have to load. It clocks in at nearly three minutes during which time the QL drives spasmotically whir on and off, which at first, might make one think something is haywire in the QL. (ASTRONOMER takes 4:40 to load from cassette.) A beutiful picture of "Earth Rise" as seen by the Apollow astronauts does come on after about 30 seconds. You will need a monitor for COSMOS, as it locks up in our TV mode with this picture on the screen. A black and white TV will work in monitor mode, but many are not adjustable enough to accomodate the entire COSMOS screen.

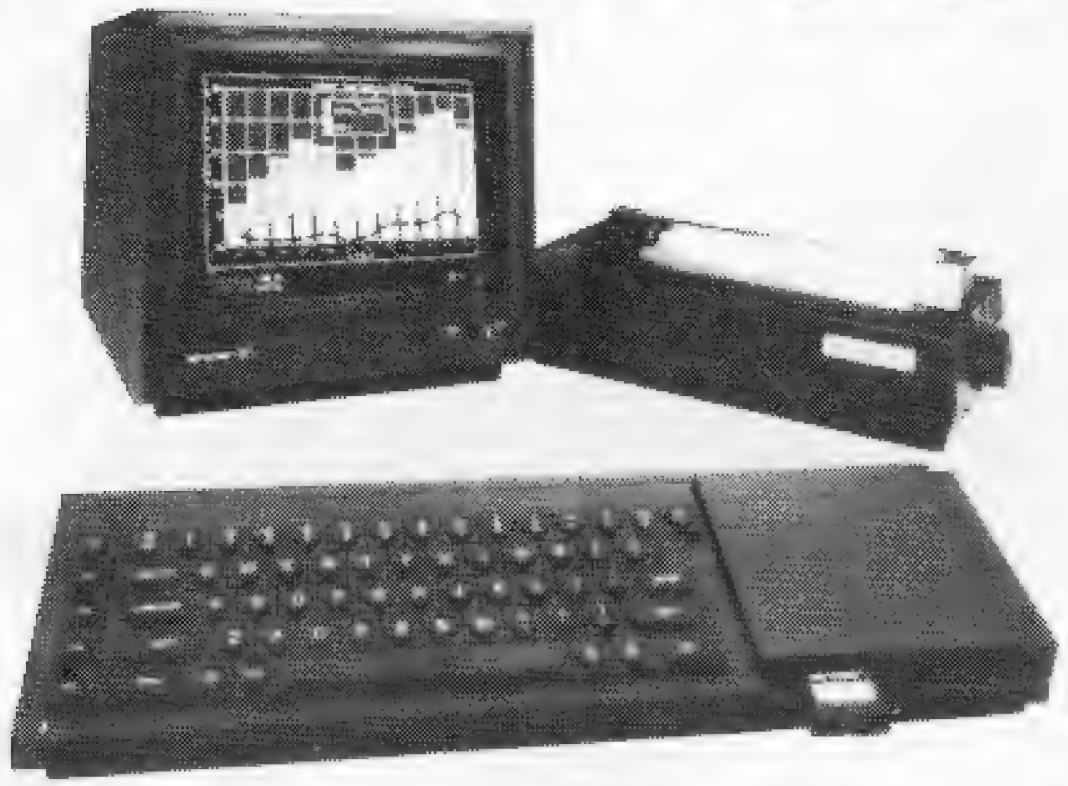
COSMOS is colorful, well laid out, and easy to read. It is user friendly with lots of superimposed menus and abilities to escape from most any option. A backup copy is required and takes 10 minutes or so. This can later be altered to include your latitude and longitude so it need not be keyed in each time, which ASTRONOMER does not allow. The select view option draws any of eight compass direction views, overhead, or a whole sky. This includes all the planets, 502 stars, moon, sun and periodically a fuzzy blob which turns out to be Halley's Comet. This takes 38 seconds--a little longer for the whole sky. A cursor option will give you a quick screen full of info on whatever object you have encircled including lots of star data. Even a graphic depiction of the planets in their phases, the moons of Jupiter, and tilt of Saturn's rings are possible. All of this uses the QL's abilities which makes it almost immediate after each key press. No constellation tracing is provided for, though. A display of the inner and outer planet's orbits is available, but no animation is possible as in ASTRONOMER.

I was horrified to discover that no printout option is available! There is a screen dump option which allows a copy of the current screen to be put on cartridge for display at another time. I am assuming that such a file could be read back to a QL printer dump program, but the scant documentation that comes with COSMOS gives no indication of how.

COSMOS is replete with security measures. The original must be present in drive 2 to load, and then any attempt to break into the program (even accidental) locks the QL up. I have been able to break into COSMOS and get a listing either on my printer or the screen of most of the files by using the QL's COPY_N command. COSMOS is almost entirely BASIC so alterations and additions may not be too tough. A printout capability would certainly be great! COSMOS is very fast once its loaded and contains much useful and interesting information. Using the QL's abilities, COSMOS does quick and accurate calculations. The ones I have checked are in good agreement with other sources. I would recommend COSMOS with the suggestion to write TALENT and see when COSMOS II will be coming out.



NOW THE SINCLAIR QL HAS OUR FULL SUPPORT.



A+Computer Response proudly announces their complete American support for the Sinclair QL and these fully authorized dealers:

- **Russell Electronics**
  Red Mill Rd.
  Center Hall, PA 16828
  814-364-1325
- **RMG Enterprises**
  1419.5 Seventh St.
  Oregon City, OR 97405
  503-655-7484
- **T.E.J. Computer Products**
  859 N. Virgil Ave.
  Los Angeles, CA 90029
  213-669-1418
- **Sharp's Inc.**
  Route 10, Box 459
  Mechanicsville, VA 23111
  804-746-1664
- **Curry Computer**
  5344 Banff Lane
  Glendale, AZ 85306
  602-978-2902
- **Foundation Systems**
  17620 26 Mile Rd.
  Washington, MI 48904
  313-781-5800
- **Sunset Electronics**
  2254 Taraval St.
  San Francisco, CA 94116
  415-665-8330
- **SCAD/ware**
  Route 16, Box 37
  Church Creek, MD 21622
  301-228-7910
- **Pyramid Electronics**
  2174 Gulf Gate Drive
  Sarasota, FL 33581
  813-922-9574
- **Harry S. Walters**
  101 Bassick Rd.
  Trumbull, Ct. 06611
- **Knighted Computers**
  707 Highland St.
  Fulton, NY 13069
  315-593-8219
- **Brice Road Pharmacy**
  1653 Brice Road
  Reynoldsburg, OH 43068
  614-861-3600
- **C.W. Associates**
  419 N. Johnson St.
  Ada, OH 45810
  (419) 634-4878
- **Info-Mation**
  RR#1 Box 260
  California, MO 65018
  (314) 796-4921
- **Variety Sales**
  325 W. Jersey St.
  Elizabeth, NJ 07202
  (201) 289-5699
- **Markel Enterprises**
  P.O. Box 2392
  Secaucus, NJ 07094
  (718) 627-0034
- **Quantum Computing**
  8 Gillen Street
  Mine Hill, NJ 07801
  (201) 328-8846

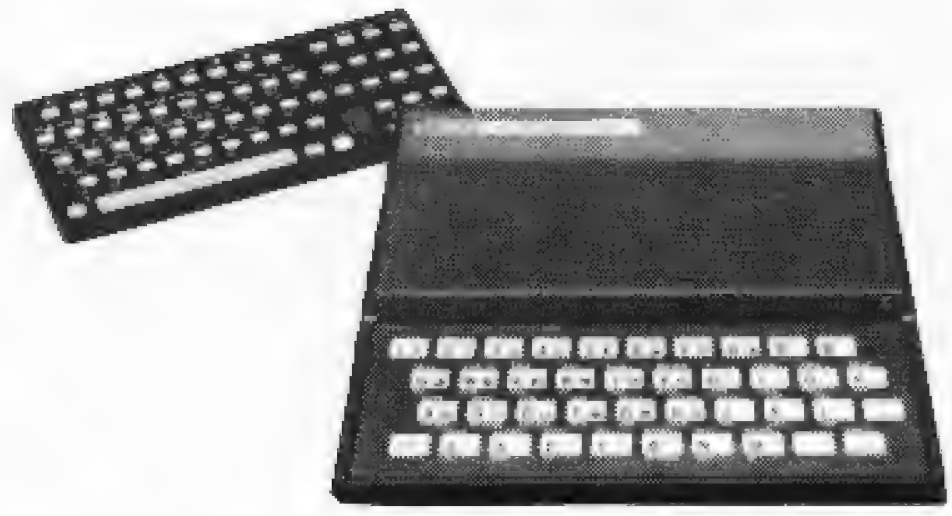
A+ Computer Response

Prompt, Professional Sales & Service

69-B Island St. • Keene, N.H. 03431 • (603) 357-1800

T/S 1000/ZX81
External Keyboard Buffer

by Tim Stoddard



If you're like me, you like to attach not only your memory pack, but also a modem, a printer, a joy-stick interface, an AC controller, and anything else that there is room for behind the computer. One of the big problems with this, especially if you have a matrix keyboard like the TI 99/4a that Radio Shack sells, attached off the main circuit board, are those "unexplained" crashes (the ones where you didn't touch any peripherals on the back) or the external keyboard seems to "lock-up" on certain keys.

It's caused by the "antenna effect". The internal or external keyboard is directly connected to the CPU address bus through diodes, and acts like an antenna; picking up all kinds of noise such as that emitted from florescent lighting.

The diodes that are used to isolate the keyboard matrix from the address bus help, but just don't do the job, especially when you add that all-important external keyboard to make your entry world a little easier. What is needed is some sort of buffer to completely isolate the address bus and provide plenty of "drive" to that nice external keyboard.

A 74HCT245 buss driver to the rescue! This driver not only provides the needed drive and isolation, but it also almost solders right in! Note that you can also use the slightly more inexpensive 74LS245, but it will use more power and dissipate more heat inside your computer.

First open your computer by removing the five screws on the bottom. When you do this, OBSERVE PROPER STATIC PRECAUTIONS! Work on an anti-static mat or sheet of aluminum foil. Keep your body in contact with the mat/foil, while handling your computer and the 74HCT245. Three of these screws are under the rubber feet. Take the back off and locate the two screws that hold the circuit board on to the top part of the case. Remove these two screws and CAREFULLY watch those two internal keyboard ribbon ribbon cables... and turn the Printed Circuit Board over.

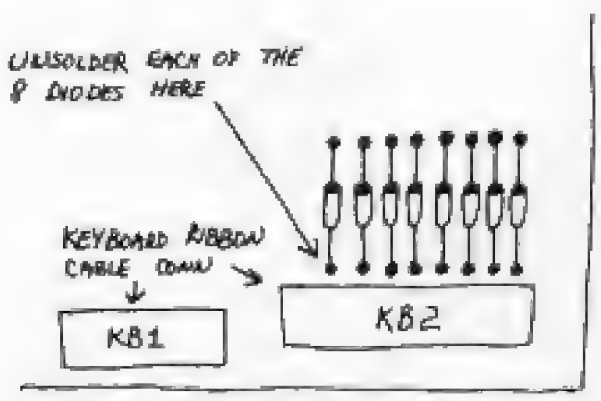


FIG 1
(COMPONANT SIDE)

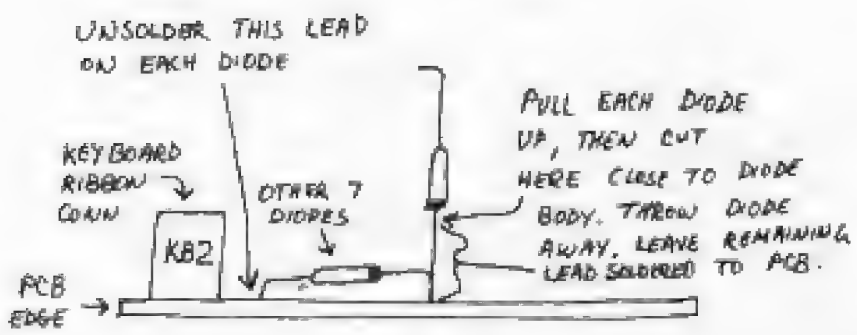


FIG 2
(PCB EDGE)

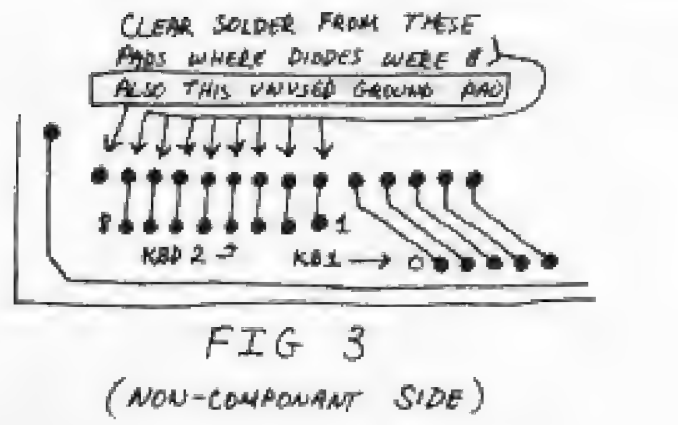


FIG 3
(NON-COMPONANT SIDE)

Using figure 1 locate the eight diodes just above the keyboard ribbon cables. Now using a pair of needle-nose pliers and a LOW WATTAGE (10 to 22 watts) soldering iron remove one side of each diode that is closest to the keyboard connector.

Using Figure 2, bend and then cut each of the eight diodes as shown.

Next turn the PCB back over and working very carefully use a solder sucker or solder wick to remove the solder out of the pads closest to the keyboard connector and also from the pad just to the left of the left-most diode location. See figure 3.

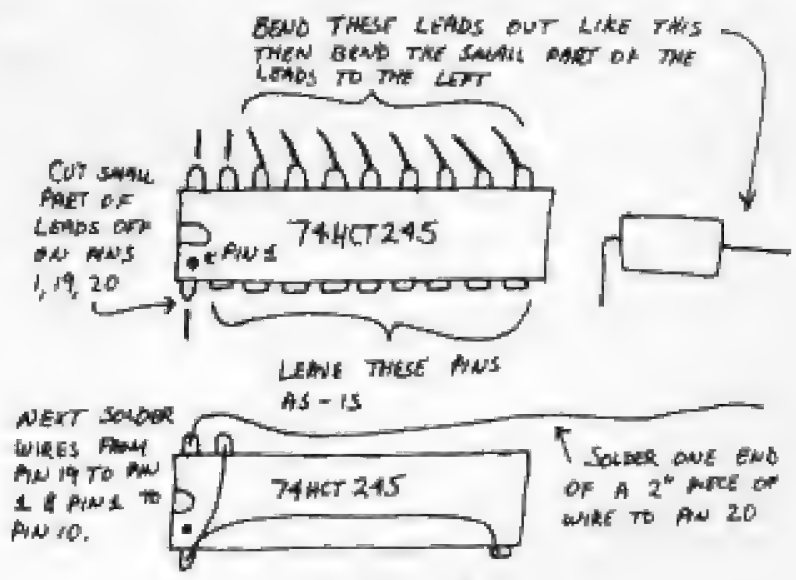


FIG 4

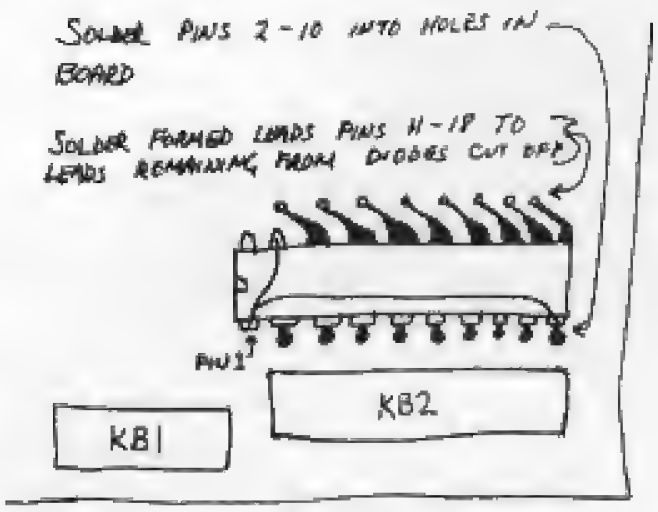


FIG 5
(COMPONANT SIDE)

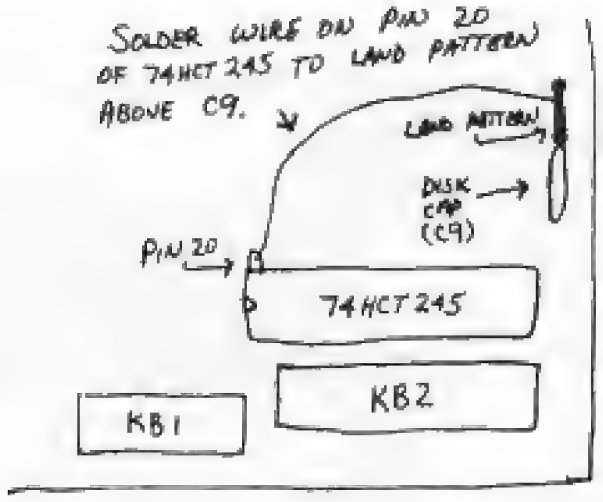


FIG 6
(COMPONANT SIDE)

Now view figure 4 and form the leads of the 74HCT-245 as shown. Then add the three wires as shown in the same figure to the IC.

Now looking at the component side of the board, insert the IC so that pin 10 goes into the pad to the right and pins 2-9, go into the pads, where the diodes used to be. Check figure 5. Once you're sure the IC is positioned correctly, solder pins 2-10 to the PCB, then solder pins 11-18 to the wire leads from the diodes.

Lastly, using figure 6, solder the wire from pin 10 of the IC to the point shown in the same figure.

Re-assemble your computer, power up and try the keyboard. If all is well, add all those peripherals and enjoy. If your keyboard only partially works, check those internal ribbon cables and insure that they are not broken.

T/S 1000
DIGITAL CLOCK

The following program was excerpted from the book: HIGHFALUTIN' COMPUTIN' by Bob Orrfelt. It is reprinted here with permission from the author. Look for more excerpts in upcoming issues. You can obtain your own copy of this book from Sunset Electronics.

Program 4-3 (from Chapter IV, which is a study on the fundamentals of Basic programming), is a Digital Clock that produces large numerals five lines high.

Lines 100-130 define the four strings needed to form parts of the numerals. [g] indicates a graphics mode space, so A$ is three black squares. Five C$'s each printed just below the last one, form a 1. These are used in the PRINT statements.

Lines 200-250 define constants needed to compact the program. Lines 2310-360 are used to set the starting time. U is the starting ten minutes digit and B becomes the units minutes digit. Lines 400-430 insert two colons at the proper position.

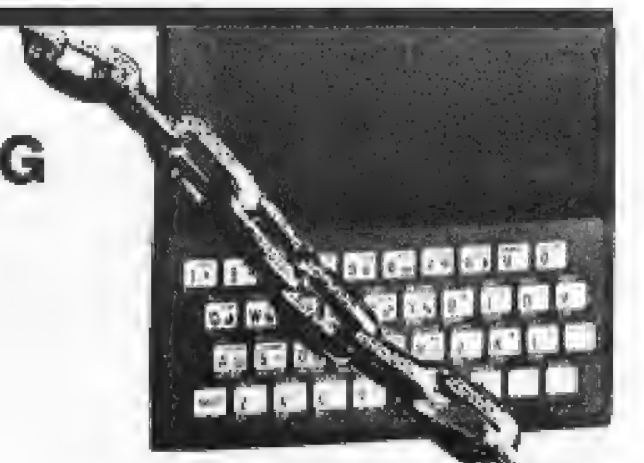
Lines 500-990 are the timing loops for the clock. The hours two digits are unique. Hours from 1 to 9 have a blank for the first digit, and only a 1 is needed for the hours of 10 through 12. Line 520 tests for the 1. If C is greater than 9; the program GOSUBs to line 2000 and inserts the 1. At 12:59:59 + one second line 970 resets the hour to 1 and clears the screen. The clock then starts off at 1:00:00. Lines 800-810 adjust the units seconds loop to one second.

Lines 1000-1095 are the PRINT statements to form the digits. THere are five ATs on each PRINT line. N, O, P, Q and R are the screen line locations. X is the column position. A$, B$, C$ and D$ are the black squares used to form the digits.

```
 10 REM "D.C"
 20 REM (C)R.ORRFELT
 30 REM AUG 1982
100 LET A$="ggg"    [g ][g ][g ]
110 LET B$="g g"    [g ][ ][g ]
120 LET C$="  g"    [ ][ ][g ]
130 LET D$="g  "    [g ][ ][ ]
200 LET N=8
210 LET O=9
220 LET P=10
230 LET Q=11
240 LET R=12
250 LET K=1000
310 PRINT " INPUT HR AND MIN"
320 INPUT A
330 INPUT B
340 CLS
350 LET U=INT (B/P)
360 LET B=B-P*U
400 PLOT 18,24
410 PLOT 18,20
420 PLOT 38,24
430 PLOT 38,20
500 FOR H=A TO R
510 LET C=H
520 IF C>O THEN GOSUB 2000
530 LET X=5
540 GOSUB K+C*P
600 FOR T=U TO 5
610 LET X=Q
620 GOSUB K+T*P
630 FOR M=B TO O
640 LET X=15
650 GOSUB K+M*P
700 FOR D=0 TO 5
710 LET X=21
720 GOSUB K+D*P
730 FOR S=0 TO O
740 LET X=25
800 FOR Z=1 TO O
810 NEXT Z
900 NEXT S
910 NEXT D
920 NEXT M
930 LET B=0
940 NEXT T
950 LET U=0
960 NEXT H
970 LET A=1
980 CLS
990 GOTO 400
1000 PRINT AT N,X;A$,AT O,X;B$,AT P,X;B$,AT Q,X;B$,AT R,X;A$
1005 RETURN
1010 PRINT AT N,X;C$,AT O,X;C$,AT P,X;C$,AT Q,X;C$,AT R,X;C$
1015 RETURN
1020 PRINT AT N,X;A$,AT O,X;C$,AT P,X;A$,AT Q,X;D$,AT R,X;A$
1025 RETURN
1030 PRINT AT N,X;A$,AT O,X;C$,AT P,X;A$,AT Q,X;C$,AT R,X;A$
1035 RETURN
1040 PRINT AT N,X;B$,AT O,X;B$,AT P,X;A$,AT Q,X;C$,AT R,X;C$
1045 RETURN
1050 PRINT AT N,X;A$,AT O,X;D$,AT P,X;A$,AT Q,X;C$,AT R,X;A$
1055 RETURN
1060 PRINT AT N,X;A$,AT O,X;D$,AT P,X;A$,AT Q,X;B$,AT R,X;A$
1065 RETURN
1070 PRINT AT N,X;A$,AT O,X;C$,AT P,X;C$,AT Q,X;C$,AT R,X;C$
1075 RETURN
1080 PRINT AT N,X;A$,AT O,X;B$,AT P,X;A$,AT Q,X;B$,AT R,X;A$
1085 RETURN
1090 PRINT AT N,X;A$,AT O,X;B$,AT P,X;A$,AT Q,X;C$,AT P,X;A$
1095 RETURN
2000 LET C=C-P
2010 LET X=1
2020 GOTO 1010
```

TS 1000/1500 PROGRAM CHAINING
Part Two

by Earl V. Dunnington



The first part of this series of articles covered: the definitions of Linking and Chaining, how to make a program self running, making one program module load another, and the VARS method of passing data from one program module to another. By data, I mean text, computed variables, arrays or strings. The second part of the series will present the System Variables and the Safe Area methods of passing data.

A complete list of all of the System Variables, their names and addresses, the number of bytes, if they are saved with the program, and if they cannot be poked without crashing the system, is in the Appendix of the User Manual. The System Variables method of passing data from one program module to another is very limited. Only the two bytes of the System Variable RAMTOP can be used for this purpose, as all of the other System Variables that can be poked without crashing the system, are reset by loading the next module. For example--type the following lines into the computer:

```
 10 SAVE "POKE"
 20 LET A=255
 30 FOR N=16477 TO 16506
 40 POKE N,A
 50 NEXT N
 60 POKE 16388,A
 70 POKE 16389,A
 80 POKE 16393,A
 90 POKE 16417,A
100 POKE 16430,A
110 POKE 16431,A
120 POKE 16434,A
130 POKE 16435,A
140 POKE 16438,A
150 POKE 16439,A
160 POKE 16507,A
170 POKE 16508,A
180 LOAD "PEEK"
```

Record the program on tape, using the command GOTO 10. When the diagonal load lines appear on the screen, after the program is saved, stop the tape and use the BREAK key to return to the programming mode. Do not rewind the tape. To clear the memory, turn off the computer. Do not use NEW as this would reset RAMTOP. Power up and type in the following lines of the second program module:

```
 10 SAVE "PEEK"
 20 PRINT "STRLEN 16430: ";PEEK
16430;", 16431: ";PEEK 16431
 30 PRINT "MEMBOT ";
 40 FOR N=16477 TO 16506
 50 PRINT N;": ";PEEK N;",";
 60 NEXT N
 70 PRINT
 80 PRINT "RAMTOP 16388: ";PEEK
16388;", 16389: ";PEEK 16389
 90 PRINT "VERSN 16393: ";PEEK
16393
 100 PRINT "NOT USED 16417: ";PE
EK 16417
 110 PRINT "SEED 16434: ";PEEK 1
6434;", 16435: ";PEEK 16435
 120 PRINT "COORDS 16438: ";PEEK
 16438;", 16439: ";PEEK 16439
 130 PRINT "NOT USED 16507: ";PE
EK 16507;", 16508: ";PEEK 16508
```

Without rewinding the tape, record the second program module using the command GOTO 10.

To run the chained program, rewind the tape. ENTER either the command: LOAD "POKE" or LOAD "" (no space between the quotes), and play the tape. As you can see, only the 255 poked into the two bytes of the System Variable RAMTOP, were passed from one program module to the next. In using the System Variable RAMTOP for this purpose, you must remember that if the value in address 16388 plus 256 times the value in address 16389 is less than 19712, a minimum Display File is set up. This may not be desirable with 16K or more RAM. If the combined value is 19712 or more, then an expanded Display File is set up, using additional memory, which may not be desirable with only 2K RAM. Also if NEW is entered, RAMTOP will be reset to the address of the combined value.

The System Variables, in which the poked value was not passed from one chained module to the next, may be poked and the value saved along with the program, like variables and strings (see "Of Strings And Things" in the March/April '86 issue of TOM). To illustrate this procedure:

Turn off the computer to clear the memory. Power up and rewind the tape. Load the "POKE" module only (by pressing BREAK when the load lines appear before the second module loads). Stop the tape. Delete lines 140 to 180 by entering only the line number. Delete lines 10 to 130 by typing in the "PEEK" program. Rewind the tape and save the program using the command GOTO 10. Turn off the computer to clear the memory. Power up, rewind the tape, and load the new unchained "PEEK" program. As you can see, the value 255 poked into these System Variables was saved except those in RAMTOP and in addresses 16477 to 16501 of MEMBOT. There are certain conditions regarding the use of the System Variables for this purpose. They are:

NAME	ADDRESS	CONDITIONS
VERSN	16393	None
NONE	16417	None
STRLEN	16430	Value must be recovered
	16431	before any LET or INPUT command in the program.
SEED	16434	Value must be recovered
	16435	before any RAND or RND command in the program.
COORDS	16438	Value must be recovered
	16439	before any PLOT or UNPLOT command in the program.
MEMBOT	16502 to 16506	Only these listed addresses can almost always be used.
NONE	16507	None
NONE	16508	Cannot be used when the System Variable RAMTOP has been poked to less than 19712 without entering NEW.

A method for determining the addresses of the Upper and Lower Limits of the Safe Area of a program was presented in "Adventures In The RAM Jungle And Other

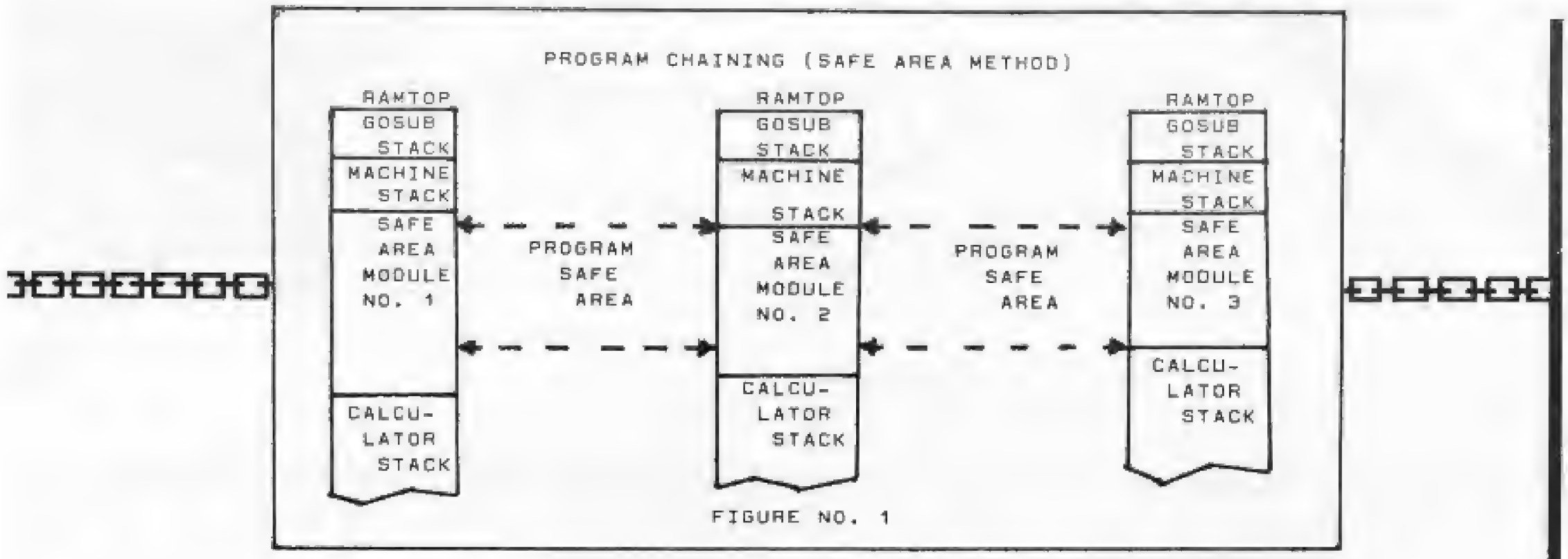


FIGURE NO. 1

Mysteries" (Sept/Oct '85 to Jan/Feb '86 issues of TDM). The Safe Area method is much more suitable for passing large amounts of data, in Chained programs, than either the VARS or the Systems Variables methods. Two disadvantages are that the data can be wiped out by NEW and by allowing the program to be listed when operating in Mode 2 or 3 (see "Adventures...", page 9, in Jan/Feb '86 issue of TDM). The amount of data that can be passed, using the Safe Area method, is limited by the program module with the lowest Upper limit of the Safe Area and the module with the highest Lower Limit of the Safe Area (see Figure No.1). The number of modules is limited only by the length of the tape and number of tapes available. Thus, when large amounts of data are to be passed, each module should be as short as possible and accomplish only one task. Combined with "Top Down" programming in each module, this is a form of "Structured Programming".

Using the Safe Area method, Z80 decimal code can be poked by the program directly into the Safe Area addresses. Each address can accept only positive values from 0 to 255. Variables with negative values or over 255 can be converted to a string, as in the VARS method, and each character converted to its code before poking. For example--type the following lines into the computer:

```
10 SAVE "POKE"
20 LET A=-12345678912345
30 LET A$=STR$ A
40 LET B=1
50 FOR N=18000 TO 18013
60 POKE N,CODE A$(B)
70 LET B=B+1
80 NEXT N
90 LOAD "PEEK"
```

Record the first module on tape using the command GOTO 10. This time you can use NEW to clean the memory before typing in the second module, as follows:

```
10 SAVE "PEEK"
20 DIM A$(14)
30 LET B=1
40 FOR N=18000 TO 18013
50 LET A$(B)=CHR$ PEEK N
60 LET B=B+1
70 NEXT N
80 LET A=VAL A$
90 PRINT A
```

Record the second module on the tape with the command GOTO 10. Rewind the tape and load and run the chained program, using the command LOAD "PEEK" or LOAD "". The display should read: -1.2345679E+13.

Beginning Z80 Machine Code

Part Three

by Syd Wyncoop

Before we get to our first MC instructions, lets take another look inside our CPU. Inside we will find registers that are called A,F,B,C,D,E,H,L,I,R,IX,IY,SP, PC,A',F',B',C',D',E',H' and L'. These are not the alphabet soup, that the CPU had for lunch. Registers are merely storage places within the CPU as opposed to external memory (ROM and RAM). Think of these registers as storage boxes with names instead of addresses...much the same as you would BASIC variables.

Some of the single registers can be married to form register pairs. You are hereby ordained, by the power invested in me by the Great God Z80, as Justice of the CPU, to form these unions as required. The permissable combinations are AF,BC,DE,HL,AF',BC',DE' and HL' (and you thought I didn't know the alphabet!).

Single registers are similar to bytes, in that they can contain any value 0-255. Register pairs can contain any value 0-65535, which makes them very valuable as address pointers. Refer to the discussion on addresses in lesson one for more on this (contact TDM if you need back copies). We will use these similarities to pass parameters (information) to and from our MC routines.

On the subject of addresses and register pairs, you need to remember which is high and low. In memory (addresses), the first byte is low, however, with register pairs, the first register is high. This is easily remembered by knowing that the HL register pair was named with this in mind. H means "high" and L means "low". An assembler will handle this for you, but we will have to watch it while we are hand assembling our code. Many crashes will occur because you forgot (or confused) the order of the high and low bytes or registers.

Some of these registers have special names and/or jobs. Chart 2 lists some of these names/jobs. However, we will not discuss them further until we get to the instructions using them.

Now for our first set of instructions (and you were wondering if I even knew any). Its mnemonic is "Ld", which is short for Load. Ld has no relation to the Basic LOAD command. Ld is an assignment instruction and acts very much like the Basic LET command.

The proper "syntax" is: Ld A,15. Which is read as "Load the A register with the value 15". Ld acts very much like the Basic LET x=15.

Take another look at the sample disassembly that I left you with last lesson. Look at the comments and see if you can follow what is happening. It is a program that will return the sum of 0Ah and 10h to Basic with the command: PRINT USR address. For practice, you can enter that program. Try poking the 2nd and 4th bytes with different values and run it again to see if you get the results you expect. If the sum is greater than 255, you will discover a bug I left (intentionally) for later correction.

Note that we loaded the result into the BC register pair before returning to Basic. This is due to the Basic Operating System's handling of the USR function. It will always return the value held in the BC register pair. The value returned will not be the result unless you properly load BC before returning.

Ld may not seem to be of much value. However, in its many forms, Ld is the "most used" instruction. We can Ld most registers, register pairs or addresses with either a constant, the contents of another register (pair), or the contents of an address. Chart 3 details some of the many forms Ld can have as well as the proper "syntax".

You will notice some instructions have parenthesis. The Parenthesis signify "the contents of". For example: read the instruction Ld A,(4000h) as Load the A register with the contents of the address 4000h. The Basic commands PEEK and POKE can be compared to these instructions. If the parenthesis appear on the left of the comma, you have a POKE operation, and if they appear on the right of the comma, you have a PEEK operation. The Basic equivalent of Ld A, (4000h) is LET x = PEEK 16384, (4000h=16384). Using this knowledge, the instruction Ld (4000h),A is equivalent to POKE 16384,x.

You will also notice a symmetry to the instructions. You can Ld r, (HL) and you can Ld (HL), r. This symmetry will prove to be very useful and holds true throughout most of the instruction set.

Note that some instructions seem to favor the register A or the register pair HL. This is due to their special functions (chart 2). There are simply some instructions that can only be performed with A or HL, and no other register (pair). We will see that Ld is not the only instruction to exhibit this favoritism. This is not as restrictive as it first sounds, although you will on occasion wish for an instruction that does not exist.

There is no need to detail the operation of each instruction as you should be able to determine appoximately what can be expected from them, if you study charts 2 and 3 in conjunction with this lesson. We will discuss them further as we use them. It will be much easier for me to explain, and easier for you to understand their operation.

Chart 2

Register	Name	Job
A	Accumulator	accumulate the results of eight bit arithmetic
		directly access the contents of any memory address
F	Flags	holds various flags for CPU which indicate the results of arithmetic and logical instructions
B		eight bit counter
BC		sixteen bit counter
DE	Destination	used for block moves
HL	High/Low	sixteen bit arithmetic
		directly access memory addresses
		indirect address pointer

Chart 3

Registers	Register Pairs
Ld r,r	Ld rr,nn
Ld r,n	Ld IX,nn
	Ld IY,nn
Ld A,(pq)	
Ld (pq),A	Ld (pq),BC
	Ld (pq),DE
Ld r,(HL)	Ld (pq),HL
Ld A,(BC)	Ld (pq),IX
Ld A,(DE)	Ld (pq),IY
Ld (HL),r	
Ld (BC),A	Ld BC,(pq)
Ld (DE),A	Ld DE,(pq)
	Ld HL,(pq)
Ld r,(IX+d)	Ld IX,(pq)
Ld r,(IY+d)	Ld IY,(pq)
Ld (IX+d),r	
Ld (IY+d),r	
Ld (HL),n	
Ld (IX+d),n	
Ld (IY+d),n	

Where: r =any single register
rr=any register pair
n =any numeric constant 0-255
nn=any numeric constant 0-65535
d =any displacement 0-255
pq=any address 0-65535

I am not listing the hex codes for all the Z80 instructions that we will use, as this is not intended to be an exhaustive study, but is meant to give you a start. The first rung of the ladder. If you have not yet obtained a good book on the subject, you can find the codes in the appendix of your Sinclair manual.

I cannot hope to give you all you will need in an article such as this. I must advise you to get a good book as a study guide and to fill in where I leave off. Rather than suggest a book that you may not like as well as I do, I would advise you to look at several. If possible, get several opinions...but get a book.

That's it for now. Next issue we will discover the math instructions. The special significance of A and HL will be very obvious after that.

$ALE

Store Hours:
10 - 6 Mon. - Fri.
10 - 5 Sat
(PST)
OUR STANDARD SALES & SHIPPING POLICIES APPLY

TS 2068 COMPUTER

INCLUDES:
TWENTY PIECES OF FREE SOFTWARE

$139.95

SUNTRONICS KEYBOARD

KEYBOARD ENCLOSURE FOR THE TS 1000 HAS SPACEBAR.

$39.95

TS 1500 COMPUTER

INCLUDES:
25 PIECES OF FREE SOFTWARE.

$69.95

QL PACKAGE

INCLUDES:
COMPUTER
MONITOR
PRINTER
I.C.E.
($946.95 if sold separately)

$795.00
(Shipping Extra)

FULL CATALOG AVAILABLE ON REQUEST.

SINCLAIR POCKET TV

2 INCH
BLACK & WHITE
POCKET T.V.

$99.00

NEW PRODUCTS

COMPLETE TS 2068 ROM DISASSEMBLY

100 PAGE MANUAL
INCLUDES EPROM PROGRAMMING INSTRUCTIONS.

$14.95

WESTRIDGE TS 2050 MODEM

UNCASED, BUT INCLUDES POWER SUPPLY AND INSTRUCTIONS.

$29.95

REMOTE JOYSTICKS FOR THE TS 2068

ATARI TYPE
REMOTE JOYSTICKS
ORIGINALLY PRICED AT $99.95

$19.95

SINGLE SHEET FEEDER

FOR THE GORILLA BANANA, PANASONIC OR ANY TRACTOR FEED-ONLY PRINTER.
(Handles up to 8½ x 14)

$9.95

Z TERM 64 CASSETTE

M TERM II NOW IN 64 COLUMN FORMAT, WITH ADDITIONAL FEATURES.
(requires OS-64 Cartridge)

$29.95

M TERM II CARTRIDGE

DOWN LOAD/UP LOAD PROGRAM FOR THE TS 2068.

$34.95

Cassette Version $19.95

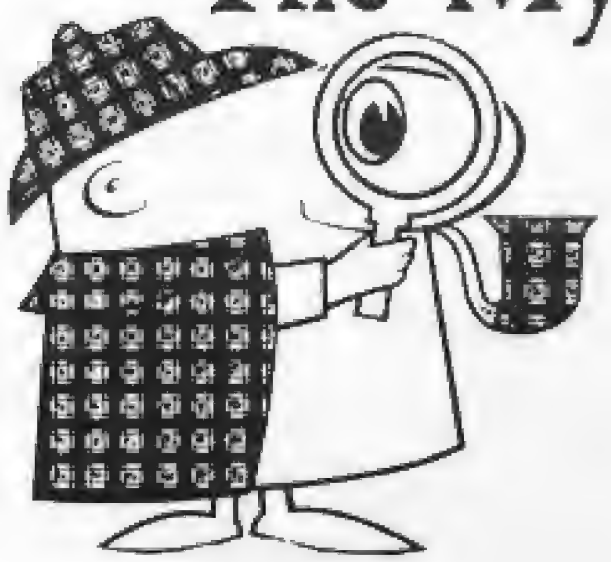
WE SERVICE ALL TIMEX/SINCLAIR PRODUCTS.

SUNSET ELECTRONICS

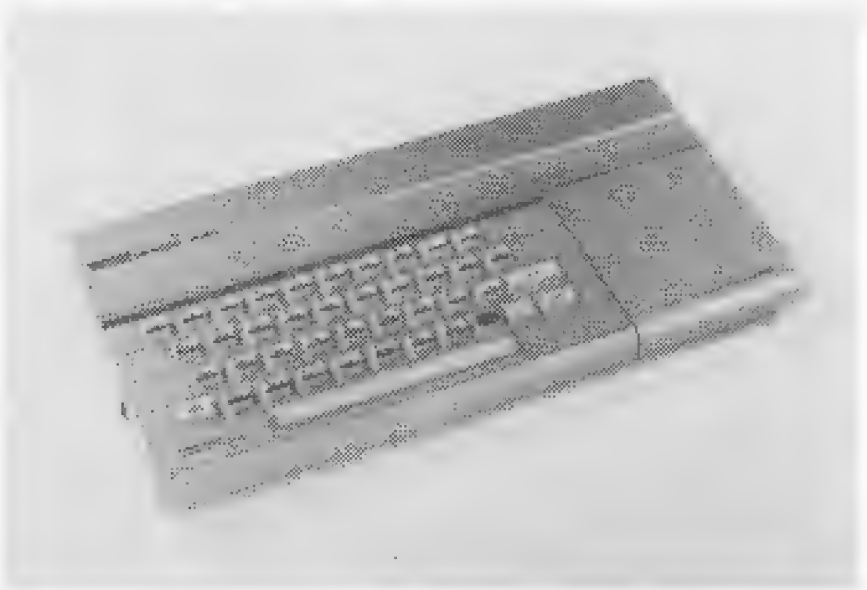
2254 TARAVAL STREET
SAN FRANCISCO, CA 94116

ORDER LINE
415-665-6161
INFORMATION
415-665-8330

"The Mystery of the Missing 253"



by Wes Brzozowski



INTRODUCTION

When Timex released its Technical Manual for the TS2068, we learned how to add memory to and switch between its three internal memory banks. This was wonderful stuff, and it's given us many excellent TS2068 products long after the "profane world" thought that the Timex computer was dead. Still, if we think back, we may remember that Timex originally promised 256 banks. In addition to the Home, Dock, and EXROM banks (which we we'll call the "standard banks"), there would also have been the provision for special enhancements that we'll refer to as "expansion banks". In this series, we will look at how those extra banks would have worked. Because the subject is very complex, we can't take up too much space with descriptions of the standard banks. That information is available elsewhere, and even without including it here, I fear that the volume of our discussions will try the patience of our dear editor. We hope you'll understand our plight.

The TS2068 Technical Manual is one very good source of information on the standard banks, and it might be a good idea to browse what it has to say on the subject. Look at pages 37-39, and 115-120, in particular. [Note: page numbers quoted in this article correspond to the original Technical Manual offered by Timex (blue cover). This same information can be found in the new second-edition manual published by Time Designs, only the information is generally a few pages earlier than those quoted.] Throughout the manual, it almost appears that the good folks at Timex tried to delete all references to the "lost 253" expansion banks. If so, they weren't entirely successful. Some pages containing copies of their internal documentation give us important clues. Both ROMs also contain code that was once intended to control these banks. Dissecting them gives a fairly clear picture of the full bank switching.

With a little "digital detective" work, we'll see that the Timex engineers planned a LOT more for bank switching than just extra memory. We'll also be able to see some of the serious problems (and clever/bizzare solutions) that graced their workbenches. Lastly, we'll see signs that they were forced to put the TS2068 into production long before it was ready.

As a result of this, our TS2068s contain large blocks of code that absolutely have not been debugged. This may well be the reason that its ROMs (and ONLY its ROMs) are socketed. Before the additional bank switching could really have worked as intended, those ROMs would have had to be replaced. This shouldn't be overly discouraging. On page 20 of the Technical Manual, we are presented with a "surgical procedure" that would allow us to replace the ROMs with EPROMs. It was quite considerate of them to include this tidbit. Perhaps they thought someone would want to debug the dormant power lying inside.

In this series, we'll use flowcharts, tables, and descriptions, to "walk through" the extended bank-switching code. Once we stop trying to figure out what it does, and instead try to understand what it SHOULD do, it's really not too hard to follow. From this, we'll also understand how the hardware of the expansion banks and Timex's unreleased Bus Expansion Unit (BEU) would have worked.

Please understand that this is a report on my own study and analysis of the Timex ROM code, and forms a self-consistent explanation of how the bank switching hardware and that code would have worked together. It is NOT a construction project. Still, it should be possible to design an expansion bank system with the information we'll be studying here. If some enterprising readers wish to correct the ROM bugs and build the necessary circuitry, I'll assist in any way I can. I give no guarantee that I've found all the bugs, but by the end, you should also have a good enough understanding of the subject to find further bugs on your own. All in all, this could be an interesting "team project".

GETTING STARTED

As you may have guessed, it will be absolutely essential that you have a TS2068 Technical Manual handy as we go through this series. Coincidentally, this very magazine can sell you a copy for just twenty-five bucks. If you've read this far, you're probably the type who'd find it useful anyway, so send 'em the money. You won't just be helping them, you'll be helping yourself. [Editor's Note: Thanks for the great plug Wes...but please tell the good folks that I didn't put you up to this!]

You'll also need some sort of disassembler. We just can't provide complete listings of the ROM code here, but we'll give TONS of memory addresses, so you can look for yourself. Ray Kingsley's excellent HOT-Z-AROS will let you look directly into the EXROM memory, which would be helpful. But if you have another version of HOT-Z, or another disassembler altogether, never fear. We need only copy the EXROM code onto a cassette, LOAD it back into some convenient RAM location, and then disassemble it. If we are clever about where we LOAD it, the difference in memory addresses will be no problem at all.

Most schemes for putting the EXROM on cassette involve a lot of convoluted bank switching and code moving in machine code. But we're going to be a little lazy, and do it the easy way. Perhaps the best kept EXROM secret is that you can do the entire job in a single line of BASIC! Just type:

```
SAVE "EXROM" CODE 0,8192
```

and you've got it! There's nothing magical about this. It just turns out that the SAVE routine is in the EXROM, and so the EXROM is already switched in whenever SAVEing is in progress. As such any attempt to SAVE from locations 0 to 8191 absolutely has to SAVE the EXROM code.

To disassemble this, you'll want to LOAD it back into RAM. If the memory addresses will be displayed in hexadecimal, then first CLEAR 32767 and then LOAD "EXROM" CODE 32768. This is location 8000 hex, so you need only subtract 8 from the most significant digit to get the true EXROM address. If you plan to disassemble in decimal, then LOAD "EXROM" CODE 4000 and just drop the 4 from the most significant digit. Note (for this second case) that if your disassembler is located below location 48192 in memory, it will overlap the code. You may want to work out some similar tricks of your own to please your particular software.

Flowchart #1 is the top level intialization routine in the EXROM. This part of the intialization was to have done all the "set up work" to find, sort out, and initialize any extra banks (RAM or ROM) that may have been added. We'll be discussing this flowchart in detail next time, but it's included here for three reasons. First, it will let the truly enthusiastic do a little extra work on their own. Second, it will help prevent later installments from getting too bogged down in flowcharts. Third and most important, it will give everyone a bit of time to practice on and get used to the notation we'll be using.

Note that each flowchart box contains the memory address of the code it represents. But the very idea of bank switching means that more than one bank of memory will be sharing the same addresses, which just begs to cause confusion. In this series, all addresses will be given in hexadecimal, but EXROM addresses will be proceeded by the letter X. As such, we can say that the NEW routine, which starts to intitialize the system variables is located at 0D1D (or 0D1D in the Home ROM) but that the routine that finishes intializing the system variables is at X096C (or 096C in the EXROM). This will save a lot of verbage, and is handy, once you get used to it.

In addition to this memory address notation, we'll also examine special shorthand ways to talk about things we'll call Bank Switching Registers and SYSCON Table Entries (we'll get around to defining these eventually). These notations have been carefully selected so as to be completely un-ambiguous, but they may require some getting used to. Also, although some users have a strong dislike for hexadecimal numbers, we hope you'll understand that they're needed here. We use numbers in a computer both for quantities and to denote various binary bit patterns. Decimal is fine for showing a quantity. But it's pretty darn hard, for example, to tell if bit 4 is set or reset in decimal 239. If we see it as hexadecimal EF, however, the experienced user can immediately see that bit 4 is zero. Since the bank switching makes considerable use of bit patterns to control different hunks of hardware, hexadecimal is the only way to go.

A Bank Switching Summary

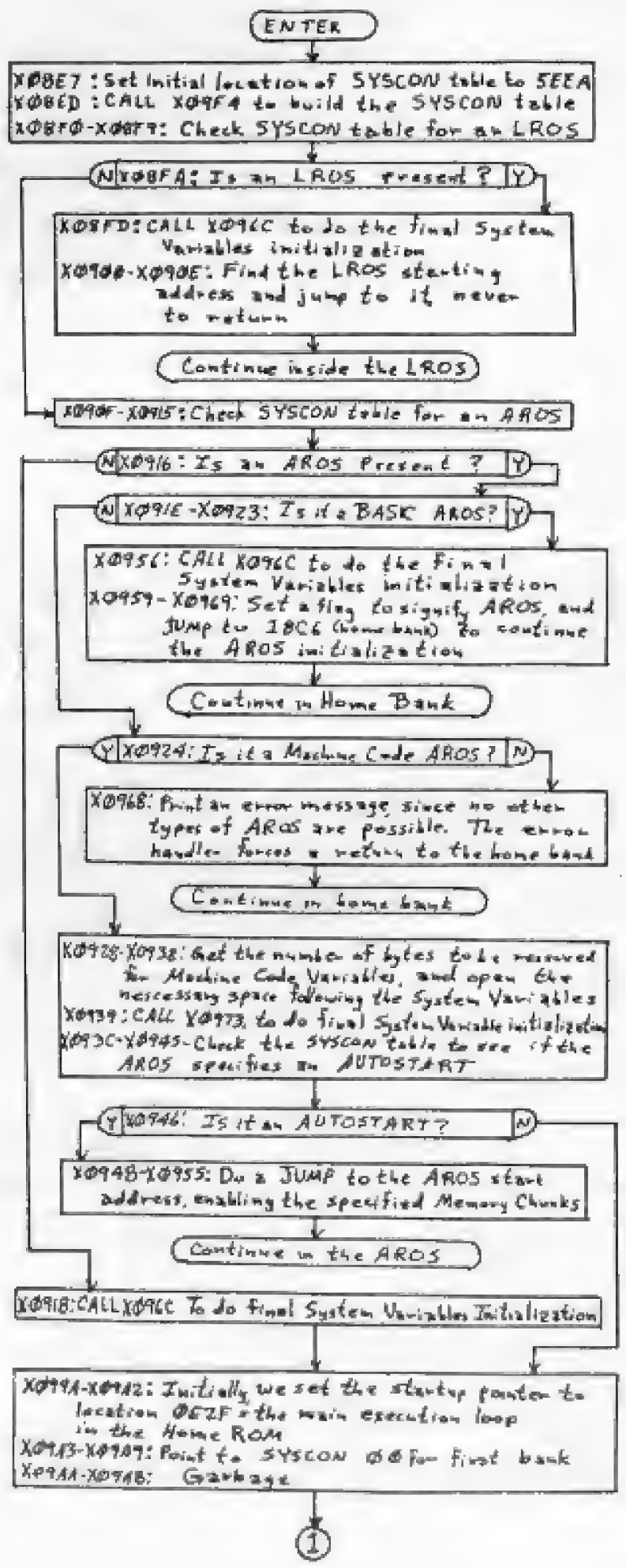
The Z-80 Microprocessor, around which our TS2068s are based, can only address 65536 (Horrors! A DECIMAL number!) bytes of memory. This is fixed in its hardware, and it's simply not negotiable. If we want it to control more memory than this, then some of the memory will have to share that "address space" in a game that's a bit like a telephone party line system. While the proposed 256 memory banks would theoretically allow control of some 16 MILLION bytes, only 65536 of them could ever be immediately available. The rest would be disconnected in a way, and waiting patiently for the Z-80 to "call them up", switch them in, and talk to them.

The TS2068 memory is broken up into 8 "chunks" of 8K apiece. They are laid out as follows:

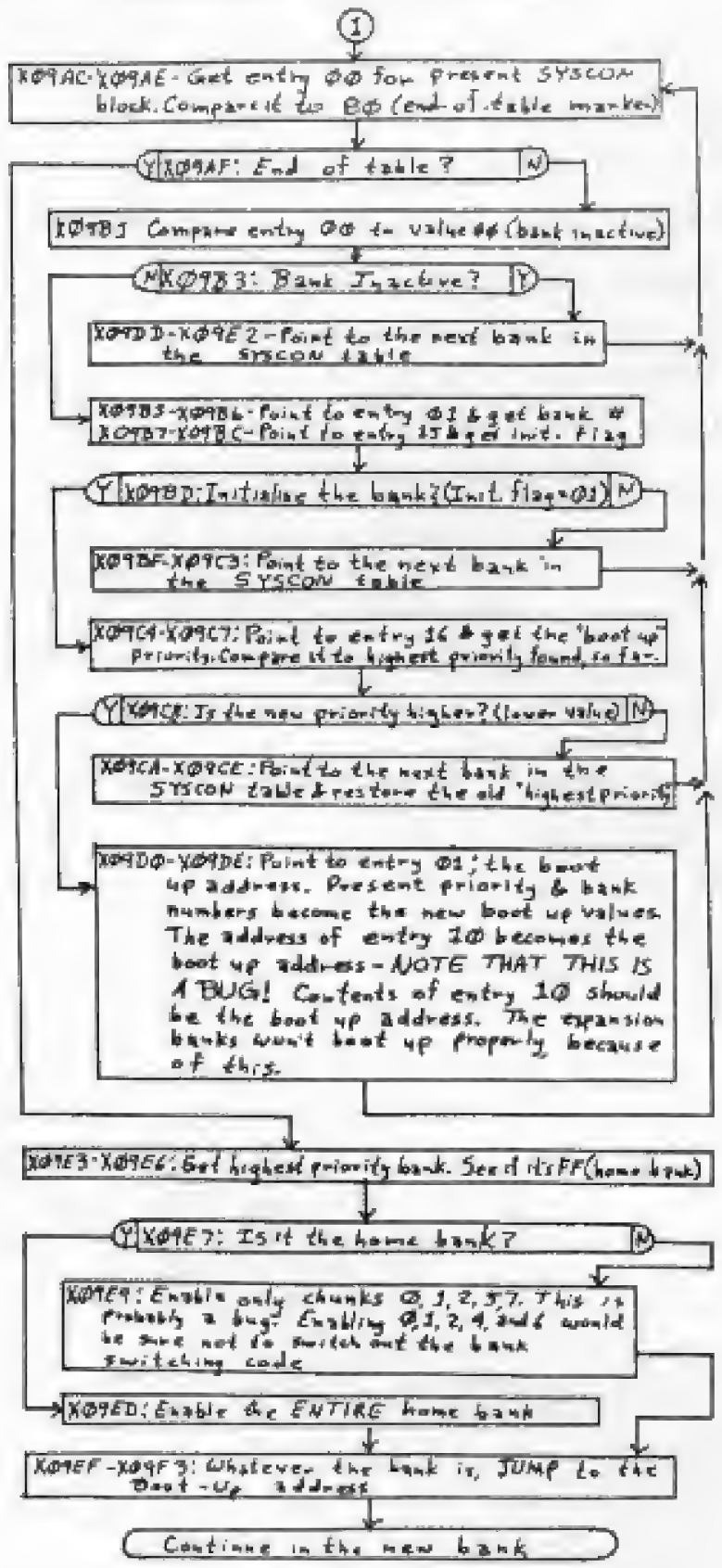
Address	Chunk #
0000-1FFF	0
2000-3FFF	1
4000-5FFF	2
6000-7FFF	3
8000-9FFF	4
A000-BFFF	5
C000-DFFF	6
E000-FFFF	7

These 8 chunks might be analogous to 8 "party line" inside the 2068, each with up to 256 subscribers. Any one of the banks (subscribers) could be using a particular line, but only 8 lines are available at a given time. Each memory bank has to have its own identifier (phone number) and it also has to have a way to know which, if any, of its chunks are presently able to

FLOWCHART 1: Top Level Initialization For Bank Switching



FLOWCHART 1 (Continued)

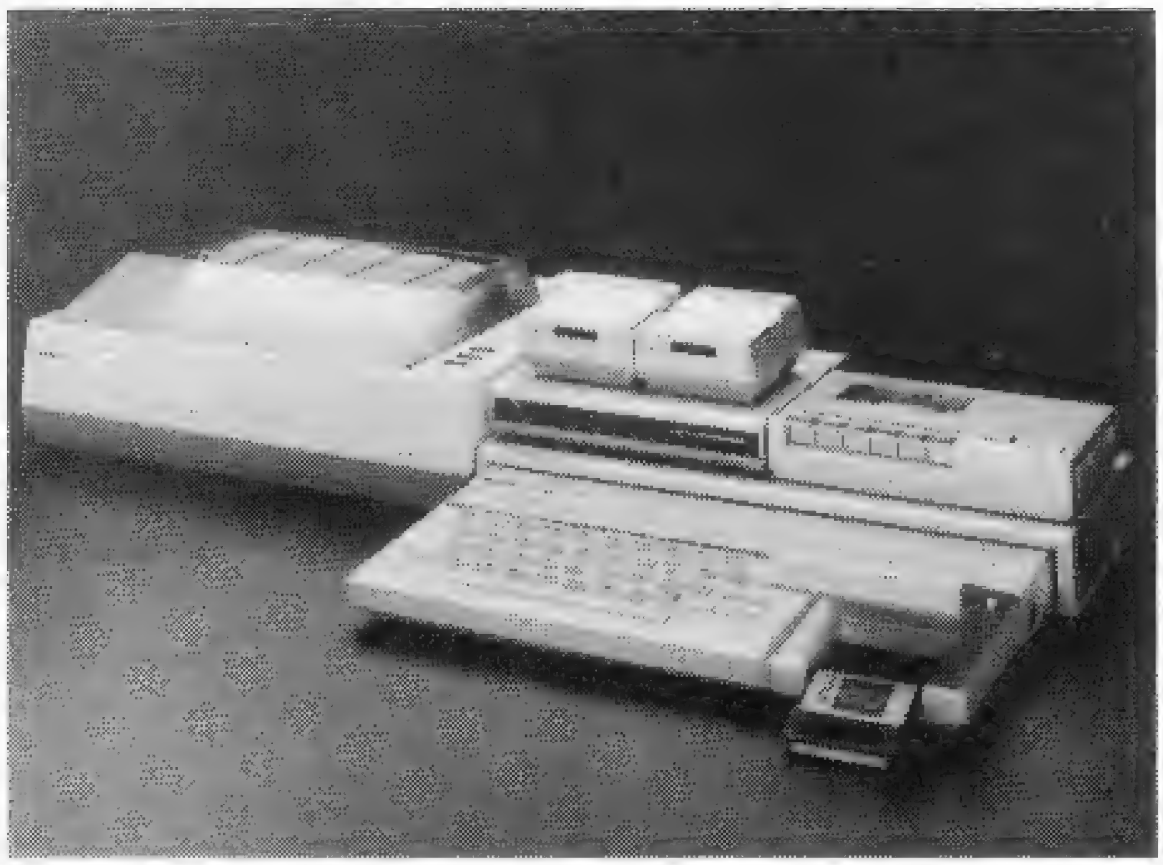


be addressed by the Z-80. This is done in what's called a Horizontal Select Register. This register contains 8 bits; one for each chunk in its memory bank. The contents of each bit tells wether its corresponding chunk is enabled (available to the Z-80). Bit 0 tells about chunk 0, bit 1 about chunk 1, and so on. If you've made sense of this, you may wonder what would happen if two banks both have the same chunk enabled at the same time. The result would be conflict, and you'd have trouble. But if you go about it properly, you can see to it that this never happens.

Now, in order to keep the TS2068's cost in line with its competition the Timex engineers put a lot of its circuitry inside a semicustom integrated circuit called the SCLD. This is a "gate array" type circuit, distantly related to the programmable arrays available to home experimenters. This type of array is programmed at the factory, however, and tends to be far more versatile and contains far more gates than the kind we might be more familiar with. The gates may be used simply as gates, or arranged into randomly ordered flip flops, allowing reasonably sophisticated functions. A disadvantage is that there is still only a fixed number of gates and signal pins to work with, and the designers may have to make some weird compromises in order to get all the functions they want out of the chip.

This fact comes back again and again to haunt the bank switching scheme.

One such example may be found in the single Horizontal Select Register used to control the three standard banks. This register is accessed through I/O port



Timex Research and Development Photo shows the never released Bus Expansion Unit (BEU) "piggy-backed" just behind the TS2068. The TS2020 Tape Recorder, TS2050 Modem and the Sinclair Microdrives are sitting on top.

F4. Although it would have seemed more reasonable to give each standard bank its own register, as is done with the expansion banks, this would have eaten up too many gates. Thus, through a "wild and whacky wisp of whimsey", the Timex engineers found a way to make do with only one; it works like this.

If a particular bit of the register is zero, then the corresponding chunk of the Home ROM is enabled. If the bit contains a one, then the corresponding chunk of either the Dock or the EXROM bank is enabled. Which one it is depends on bit 7 of I/O port FF, which, I suspect, just happened to be left over, with nothing important to do. If this bit contains a one, then it's the EXROM bank; otherwise the Dock bank applies. As we said, we can't really take too much space to discuss this. The capsule description given here is just included for completeness. A more complete description for switching the three standard banks may be found in the TS2068 Tech Manual.

Leaving the "nuts and bolts" of the standard banks behind, we should still examine some of it's consequences. The one of most immediate importance is that this scheme prevents your having chunks from both the Dock and EXROM banks enabled at the same time. Ordinarily, this wouldn't have been important. The EXROM bank was only intented to "catch the overflow" of the code that couldn't have fit in the Home ROM. The Dock bank was intended only for cartridge based software. If any other banks were needed, well, there was space for 253 more, right? Unfortunately, those banks never became available, and ingenious TS2068 users have had to use these three as best as possible.

But this minor perversion has its problems; it's important that you exercise care in trying to access the EXROM while running in the Dock bank. But there are even more subtle ways that this little foible can trip you up. In articles I've written on running RAM in the Dock bank, I've always cautioned the readers not to try to LOAD anything directly into the Dock bank. The proper procedure is to LOAD the code into the Home Bank and then transfer it yourself. I've always shyed away from explaining exactly why this is so, but having gone through this long explanation, it now can be told! The fact is, the LOAD routine is in the EXROM, and so while your're LOADing, none of the Dock chunks can be enabled. (Remember, you can't have EXROM and Dock chunks enabled at the same time.)

As such, any attempt to LOAD data into the Dock bank will instead cause the TS2068 to try to LOAD the data into the EXROM bank, where there's no RAM to be

had. Furthermore, you can't put RAM into the EXROM bank without messing with your TS2068's innards. The 8K ROM in that bank is mapped into all 8 chunks of that bank, due to incomplete address decoding. You just can't win.

The odd use of one Horizontal Select Register to control three banks has another consequence. The Home bank always "assumes" it's enabled, unless told that the Dock or EXROM have a particular chunk. This ordinarily leaves no way for the other 253 banks to be enabled without conflicting with the Home Bank. This is dealt with in a "cheap and dirty" manner with a signal on the TS2068 rear connector, called $\overline{BE}$. When this signal is low however, all internal memory is disabled, no matter what the Horizontal Select Register for the standard banks says. This would allow the additional expansion banks to "muscle their way in" when it's their turn to "talk".

The TS2068 appears designed to contain almost no circuitry that would support the expansion bank switching. That would be contained almost entirely in the never-released (and possibly never built) BEU, and the expansion banks themselves. An early map of TS2068 I/O port assignments shows ports FC and FD reserved for bank switching. For good reasons, to be discussed later, this is not the way it's turned out. These ports are never used in either ROMs, and communication with the bank switching circuitry is instead done through a memory mapped register scheme.

Four Bank Switching Registers are used. We will call them registers C0, A0, 80, and 40. These are the ways that the bank switching software refers to them, so it will make it easier to follow. Also, it's useful to retain the second digit even though it's always zero. This will prevent registers A0 and C0 from being confused with the A and C hardware registers inside the Z-80 itself. When we write a value to these registers, we are sending bank switching information to the (presently non-existent) BEU and expansion banks. However, when we read the registers we get back different information relating to the status of various banks. WE DO NOT GET BACK THE SAME INFORMATION WE SENT. Furthermore, although we send out 8-bit groups of information, we read back only 4-bit groups. That is, only the low nybble of the byte contains useful information. A summary of the Bank Switching Registers follows, and we'll explain them in detail next time:

Register Purpose

Outputs:

40 Horizontal Select. Receives the horizontal select byte (hi-active) for the "presently accessed bank"

80 Bank Number Access. Sets the "presently accessed bank"

A0 In Setup Mode: Receives the assigned bank number for the bank presently selected by the daisy chain
In Normal Mode: Receives the universal deselect byte. Chunks are hi-active

C0 Command Register. Four commands have been found:
00-Reset daisy chain & enter the setup mode
01-Step the daisy chain to the next bank
02-Reset the nybble steering logic
04-End the setup mode & enter the normal mode
-Note that no more than one bit is ever set simultaneously
-The hardware of this "register" must be able to accept the 02 command, wether it's sent as one or two nybbles, and it must be able to properly interperet the command, even if the nybble synchronization is faulty.

Inputs:

40 Least significant nybble-Horizontal select for "presently accessed bank"

80 Most significant nybble -- for register 40

A0 Least significant nybble-bank status for "presently accessed bank"

C0 Most significant nybble -- for register A0

These Bank Switching Registers are intended to control all banks EXCEPT the three "standard" banks. Each bank has a number to identify it. For the expansion banks, these are defined through the intialization software. If seven expansion banks wre present, for example, the banks would be numbered 01 through 07. Additional numbers are allocated as needed. The three standard banks, on the other hand, have fixed numbers:

Numbers For Standard Banks

FE - EXROM Bank

FF - Home Bank

00 - Dock Bank

Now suppose we wanted to read from or write to one of the Bank Switching Registers. The software for it is already in place when you turn on your computer. Appendix A of the TS2068 Technical Manual has the assembly code listings for the RAM resident code, which includes the routines WRITE_BS_REG (write to Bank Switching Register). After a short description, we'll look them over, and see how they work.

The WRITE_BS_REG routine at location 635C will write the value in the E register to the Bank Switching Register whose number is in the D register. To do this we first make a memory address out of the value of the Bank Switching Register. The register value becomes the two most significant hex digits, and the other two digits are zeros. For example, register A0 becomes memory address A000.

Eventually, we'll be writing our data to this memory address, and the BEU or a bank will pick it up and put it in the proper register. But how will the bank "know" that we're talking to it, and not just trying to use that memory location for some more mundane purpose? Another signal has to be sent out, to indicate wether the memory write operation is intended for memory or for the bank switching. This normally unused signal is on the rear connector, and is called IOA5. This signal comes from the sound chip, of all places, and is one bit of an I/O port it contains.

With IOA5 low, the data written to certain memory locations (A000, in this example) will also get written

to a bank switching register (the A0 register, in this case). For reasons to be explained later, we only write four bits at a time. That is, only the four least significant bits are accepted by the register. The first memory-write sends the low nybble, and the second write sends the high nybble. Since it's possible for a glitch to cause the hardware to "lose sync" and try to accept the high order nybble first, a "reset" to steer the nybbles properly must also be sent out. After this occurs, the hardware is set to accept the low order nybble next.

Some readers may be amazed that a mere 81 bytes can make such a complicated subroutine! This does sometimes happen when a function is divided partway between hardware and software, and here's a prime example. The fact that sanity was sacrificed for a low cost design doesn't help, either. For now, it would be helpful to review the section in the TS2068 Technical Manual on the registers in the sound chip (pages 21 and 22). Then we'll go on and look at the actual subroutine.

Ready? OK, here we go! Turn to Appendix A of the Tech Manual, and look at location 635C. Here's a blow-by-blow description of what's happening:

635C-635E - Saves the registers (so far so good).
6360-6367 - Saves the contents of the memory locations we're going to wipe out in a moment. Location C000 always takes a hit. Also wiped is the memory location that corresponds to the registe we're going to write to. (For register A0, this is location A000.)
6368-6375 - Saves the contents of the sound chip registers we're about to wipe out.
6376-637D - Sets the sound chip I/O port to OUTPUT mode.
637E-6384 - Sends 00 to the sound chip output port. This will clear IOA5, on the rear edge connector.
6385-6389 - Now that IOA5 is low, this causes the low nybble of 02 to be sent to Bank Switching Register C0. This resets the nybble steering logic, so that the next nybble written out will be accepted as the low order nybble. Note that the C0 register is only receiving a single nybble, in this case.
638A-638B - Finally! We're sending the low order nybble to the Bank Switching Register we want to talk to.
638C-6393 - Shifts the high order nybble into the four least significant bits, so it can be sent out.
6394 - Sends out the second nybble.
6395-63A2 - Puts the sound chip registers back the way they were. As such, IOA5 goes high again.
63A3-63A8 - Restores the memory locations we wrote over. Since IOA5 is now high, this does NOT write new values to the BEU.
63A9-63AB - Restore the registers we changed
63AC - ...and RETurn to the CALLing routine with everything exactly as it was, except that a Bank Switching Register has changed!

If you've gotten this far, congratulations. But you may want to get yourself a cup of tea, coffee, or whatever more potent nerve settling beverage you'd like. We're about to do the same thing with the READ_BS_REG routine!

While we write to the Bank Switching Registers one nybble at a time, there still 8 bits wide. When we read them, however, they're only four bits wide. (As we said before, we don't read back the same information we've written.) Because of this, we have to read two registers to get enough information to fill a single byte.

The READ_BS_REG routine at location 63AD reads a nybble from the Bank Switching Register whose number is in D, and another from the register whose number is in E. It then packs them both into the E register. Here's how:

63AD-63AF - Save registers.
63B0-63B2 - The programmer was probably copying code directly from the WRITE_BS_REG routine. This portion is useless here.
63B3-63B6 - Save the contents of C000, before we use them.
63B7-63B8 - More useless code.
63B9-63C7 - Save contents of two sound chip registers about to be wiped.
63C8-63CF - Set sound chip I/O port to OUTPUT mode.
63D0-63D6 - Send 00 to I/O port so IOA5 goes low.
63D7-63DB - Reset nybble steering logic.
63DC-63DF - Register (D) is read, and the useful information from it is put into the least significant nybble of C.
63E0-63E9 - Register (E) is read, and the useful information from it is put into the most significant nybble of A.
63EA-63EB - Both nybbles are packed into E.
63EC-63FA - Restore original sound chip registers.
63FB-63FC - More useless code.
63FD-6400 - Replace the contents of location C000.
6401-6403 - Replace registers.
6404 - RETurn (at last!!!).

Note that the code we refer to as "useless" is not at all benign. The three parts hold each other in check, counteracting each other, and making it appear that all three parts don't exist. But if we remove some but not all of them, the remaining part(s) will cause all sorts of mischief. So if you wish to modify this routine, beware!

These two routines form the lowest level interface between the rest of the bank switching software and the actual hardware. From here on, we'll just set the Z-80 registers up to read or write to a particular Bank Switching Register, and CALL the appropriate routine. We needn't worry about how it's done. That is, unless it's desired to experiment with bank switching hardware; then the knowledge is absolutely fundamental.

Some readers may look at these two incredibly convoluted subroutines, look back at the earlier statement that the bank switching software isn't too hard to follow, and then wonder wether my brain hasn't dropped a bit or two, somewhere. Please be assured that the rest of the bank switching code is much more civilized, however comical it may become. If you've come this far, I beg you to read on.

This discussion will generate a lot of questions. Probably the first and foremost arises from the very idea of reading and writing nybbles to memory mapped I/O, and that question is simply, "Why?" Once again, the use of SCLD gate arrays for a cheap design comes in and messes up the bank switching scheme.

Our good friends at Timex could have made things much simpler for us. They could have used I/O ports FC and FD to control the Bank Switching Registers in a manner similar to the two-port scheme used on the sound chip. This would have reduced the two subroutines we've discussed to a few simple instructions, and we could have sent 8 bit information back and forth, as well. The circuitry would be simpler, and easier to follow. All we'd have to do was run a few more signals to it.

That last sentence is the killer that sends chills through the hearts of every chip designer. Though we rarely think of it, each chip has only a limited number of pins. The more complex the chip is, the harder it is to get all the signals you need in and out of the package!

Each expansion bank would likely have contained its

own SCLD, to hold the registers for that bank, and do its bank switching chores. It's limited pinout is the probable cause of the problem. By writing one nybble at a time, only four of the 8 Data lines (D0-D3) would have to be run into the chip. By using memory mapped I/O, the signal IORQ would not be needed by the SCLD.

We've already eliminated 5 pins, and that makes any chip designer smile. A possible 6th pin would also have been saved if the designers intended to make the maskable interrupt (a subject we won't cover here) available for general use. In some cases, the MI line would then be needed to distinguish the difference between an interrupt service and a true I/O request.

Now, 5 or 6 pins is a lot, even if we've got 40 to work with. Actually, a preliminary circuit design suggests that a RAM bank SCLD would need only 28 pins, and a ROM bank only 20 pins. These are all standard pin groupings, and the lower the number you can get away with, the cheaper your design. And in the cutthroat atmosphere of the computer business, EVERY penny counts.

Sadly, since we can't put lots of functions on a single chip, this offbeat switching scheme simply gets in our way. Note that the READ_BS_REG and WRITE_BS_REG routines do essentially all communication with the Bank Switching Registers. (One renegade routine tries--and misses--communicating with the registers; this can be ignored.) As such, it might be worthwhile to consider rewriting those two routines to use I/O ports FC and FD, instead. Perhaps we could write the register number to port FC, and read or write our data from port FD. This would do a lot to simplify the Bank Switching hardware.

Should anyone want to experiment a bit with building Bank Switching Registers as Timex envisioned them, note that only address lines A13-A15 need to be tested to see if a register is being accessed. This will simplify your circuitry. Note that only some of the Bank Switching Registers are really true registers. Others will serve to reset only selected bits of a different register, and others switch hardware modes without being "stored" in any register at all. (Register C0, bit 1 simply clocks a shift register, for example.) We'll explain it all next time, but this is mentioned so that no one gets too serious about designing a bank switching system until we go over a few more things.

Unfortunately, this article's volume has already expanded beyond all pretentions of sanity, and we have not even covered all of the basics yet. I must apologize for the somewhat sketchy treatment of some topics. I've been hounded for some 18 months to get this information into print, and I've tried to include as much sheer information as I could, to appease some of those who are already familiar with the code in the EXROM.

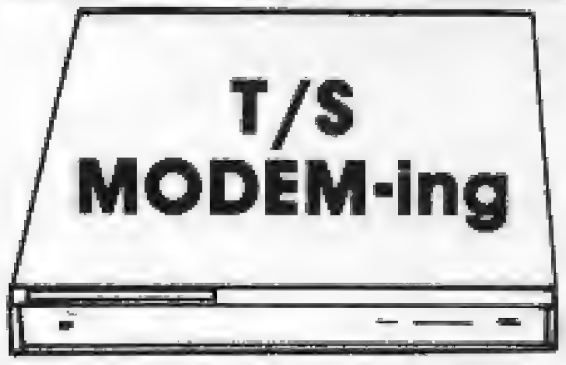
For the rest of you, I'll be filling in the blanks next time, particularly on the Bank Switching Registers. We'll also look at the RAM resident code, the SYSCON table, and the daisy-chaining of the expansion banks. Doing all this, we'll finally start looking at how the TS2068 handles it all with a guided tour of Flowchart 1, which is included here.

HOMEWORK

For those readers who don't want to wait two months to learn more, I've left lots for you to do on your own. Read the short explanation on the System Configuration (SYSCON) Table on page 81 of the Technical Manual. Put the EXROM on cassette, and compare the disassembly to Flowchart 1. Use this to begin your own annotated disassembly of the bank switching code. My SYSCON notation (to be explained next time) needs a quickie explanation to do this. The phrase "SYSCON 00" refers to the first entry of a 24-byte block associated with an EXPANSION bank. We'll pretty much ignore the AROS and LROS parts. Try to wade through the listings of the RAM resident code in Appendix A of the Technical Manual. Read the comments, and try to understand what the various routines do. In short, there's plenty to keep you busy for two months.

I'd like this series to be an interactive one. If you're particularly interested in certain things, or need more detail, let me know. Future columns could very easily cover them. If you disagree with anything I've said, or think I've missed something important, PLEASE let me know. Also, feel free to write or call with questions. I'm Wes Brzozowski, 337 Janice St. Endicott, New York 13760. If you want a reply, please enclose a stamped, self-addressed envelope. If you're in a hurry, don't be afraid to call at (607) 785-7007. I'm very friendly, provided you don't call collect and call BEFORE 9:30 PM, EASTERN time. Hope to hear from you!

Editor's note: Wes Brzozowski is an electrical engineer by profession, and is employed by an international computer giant. Wes is a member of the SINCUS T/S Group in New York, and a regular columnist for the group's newsletter.



Embellishing MTERM II

by Michael E. Carver

As editor of the Portland Area Timex/Sinclair Users Group (P.A.T.S.) newsletter, I wanted to upload the newsletter to members who own a modem, MTERM II, and Tasword. (NOTE: Uploading means sending data from one computer to another. Downloading is receiving data from another computer.) But I did not want to man the phone lines, waiting for incoming calls. The solution? Modify MTERM II to accept a remote command, from the caller, to dump the newsletter (which is stored as a Tasword textfile in the buffer). But how? Overwrite portions of MTERM II handling the remote Buffer, Start and Stop commands. (I intentionally overwrote the Buffer Open and Close routines, to avoid accidental corruption of the Buffer.) With the modifications installed, I was able to boot-up MTERM II, plug my modem into the phone line and let the computer do all the work. The following will allow one to post messages, letters, bulletins, ect. for remote downloading without the need for an operator being present.

Key in Listing 1, then load the MTERM II code. When MTERM has loaded, RUN the BASIC program. The new code will be installed in its proper location within MTERM II and you will receive prompts to SAVE this modified MTERM (Please note that this modified version will ignore any Buffer commands from the caller).

Now that the MTERM modifications are complete, we need to make a few changes in Tasword II. These changes will embed special MTERM commands before and after the text, saving them to tape along with the text. To make downloading easy on the caller, I wanted my computer to send an Open Buffer command before transmitting the text and a Close Buffer command after complete transmission. Load Tasword II, go to the Menu to get into BASIC. Edit

lines 1030, 1040 and 1100 to match Listing 2. These changes will provide a Control R at the start of the MTERM Buffer, which will automatically open the caller's Buffer. These modifications will allow the use of Tasword for typing and editing a message base for the MTERM Buffer. After you have edited Tasword with Listing 2, ENTER as a direct command [CLS GO TO 20]. This will bring you back to the menu. Make a copy of this new version by choosing option [T].

We are now ready to test out our new modifications. Prepare a test text using the modified Tasword. (Note: Texts which have been created in an unmodified Tasword can be loaded into the modified version. When the text is saved from the modified version, all changes will be made to affect remote downloading.) Save the text to tape, load MTERM II, then load your text file into the Buffer (I use Loader IV). I like to use the following parameters in MTERM to upload/download text files:

WORD = 8 STOP = 1 PARITY = None
DUPLEX = Half CONVERSION = None

Have a friend call you with their MTERM Buffer empty and closed. Upon a connect, they should send a CTRL Q ([CAPS SHIFT/7] [Q]). This command will automatically trigger your computer to open the caller's Buffer and upload your text file. Sit back and watch the text fly by. After completion, the program will close the caller's Buffer. When the caller hangs-up, your computer automatically resets itself and waits for the next caller. All of this was done whithout your having to lift a finger or press a key.

TIP: I like to set up the macro-keys (starting with "0" and chaining them together) with an introductory message, telling the caller to send a Control Q to receive the text. When the caller makes their connect, they simply send a Control E ([CAPS SHIFT/7] [E]) which automatically sends the macro-key "0" message and any others chained to it. A closing message can also be entered while in MTERM. By opening your Buffer and returning to Terminal Mode, any message can be appended to the Buffer. (Note: Be sure to close Buffer before going on-line.) As this text will follow the embedded CTRL T, it will only be sent to the caller's screen and not their Buffer.

LISTING 1

```
1000 FOR X=54211 TO 54251: READ
Y: POKE X,Y: NEXT X

1010 FOR X=54252 TO 54262: POKE
X,0: NEXT X

1020 PRINT "Saving MTERM IIa": S
AVE "MTERM IIa"CODE 54016,9216:
CLS : PRINT "Rewind Tape and Pla
y to Verify": VERIFY "MTERM IIa"
CODE 54016,9216

1030 DATA 32,31,33,0,241,203,182
,42,83,92,34,232,238,34,234,238

1040 DATA 42,75,92,34,236,238,25
4,19,40,5,33,0,241,203,246,175

1050 DATA 201,254,19,32,17,24,21
9,175,201
```

LISTING 2

```
1030 LET I=VAL "12": GO SUB VAL
"800": POKE (b-VAL "11"),VAL "18
": POKE (b+a),VAL "20": SAVE a$C
ODE b-VAL "11",a+VAL "12": CLS

1040 PRINT AT VAL "8",VAL "0";"t
ext file ";a$;" saved:";AT VAL "
10",VAL "0";a+VAL "12";" bytes,"
,a/PEEK VAL "62237";" lines"

1100 CLS : GO SUB VAL "900": VER
IFY a$CODE b-VAL "11",a+VAL "12"
```

Ultra-Easy Designer Graphics

by Paul Bingham

When I first ordered my 2068 back in, well, what seems eons ago, one thing I looked forward to most was the machine's touted User Defined Graphics capabilities. What a boon to an old ZX81 programmer this would be! On the face of it, UDGs looked pretty exciting. No matter that there were only 21 of them, ruling out easy trial of new fonts and type faces--just having any would be great.

Oh, but the real rain on the parade came when I read through the manual's five page chapter on UDGs. It became apparent that with POKE USER "a" + whatever and countless BIN sequences needed to set up each and every savory UDG byte, these little jewels were going to be bears to work with. Now as is usually the case, things were not as hopeless as they seemed after a little investigating. As it turned out, the UDG area is just a section high in memory where there is room for 21 segments each eight bytes long. And to my surprise, I did not find my ones and zeros I had POKEd, but rather decimal numbers between 0 and 255. Somewhere along the line, the 2068 was converting the barrage of zeros and ones to elegant code. If one could find a way to POKE the right codes in directly then the POKE USER BIN nonsense could be circumvented...

So for all who have wanted to exploit the use of 2068 UDGs easily, I have written a colorful little 6K utility which leaves you all the fun of designing and little of the drudgery. As can be seen in Figure A, it allows the user to design four UDG characters on screen at once and by the use of on-screen menus to alter, move or save them. All features are accessed by the top row of keys and ENTER as shown in Figure B.

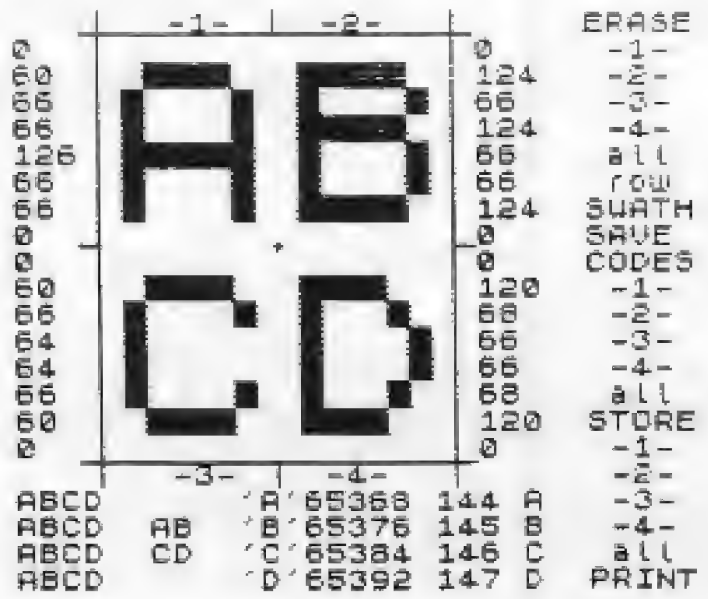


Figure A

KEY DESCRIPTIONS	
Key	Operation
1	acts like ENTER for menu selection
3	menu cursor down
4	menu cursor up
5	sketch cursor left
6	sketch cursor up
7	sketch cursor down
8	sketch cursor right
9	darkens pixel
0	deletes pixel

Figure B

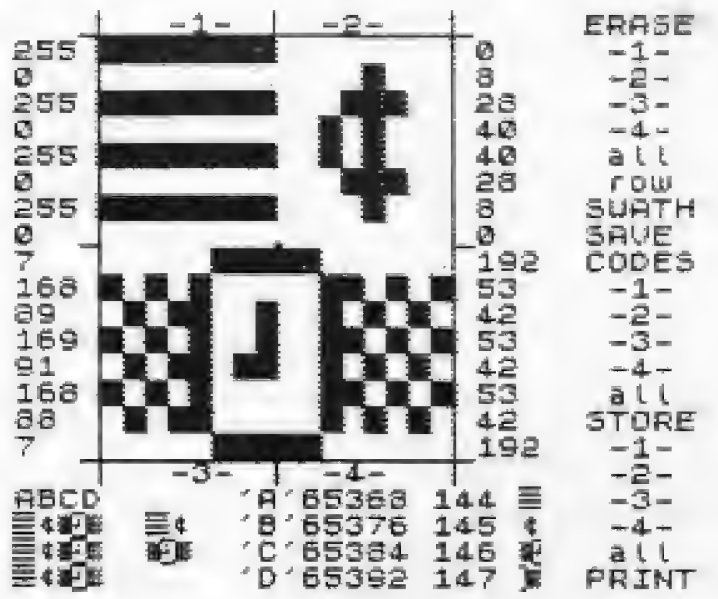


Figure C

Creating new symbols and characters is as simple as moving the cursor. Figure C shows a watch that is actually two UDG characters side by side. This shows that one need not be bound by an "eight X eight" block, but that multiple character designs are possible.

In the 2068 manual the UDG characters are referred to as "a", "b", "c", ect. to "u". We will also refer to them as such in some parts of the program. In other places they are referred to by character codes. By telling our 2068 to PRINT CHR$ 144 it will print UDG character "a" because that's its code! CHR$ 164 will get you UDG character "u", which is the last one. The program keeps you reminded of the current letter and code along with the memory location of each UDG at the bottom of the screen as can be seen in Figures A and C. You might read the manual's pages 197, 198, and 243 also.

Now about the listing. The program will serve you without lots of documental aids. So it will suffice to get you on your way with just a few tips:

In trouble by hitting the wrong key? Just hit break or if you are in an Input sequence, type in the command STOP. As long as the designs you have been working on are in the UDG area, you can re-RUN without losing any of them. GOTO 100 works just the same and won't harm any variables.

Be sure to call for "CODES" on the menu for all the blocks you have designed on the screen BEFORE calling for "STORE" on the menu. Without the codes, the "STORE" feature will store blanks (zeros) in the UDG area. STORE sends the particular codes for the block (or all four blocks if you chose "all") to the letter destination you input. Inputting "a" puts it in the first position. But you can send them anywhere or even to multiple locations.

Now if you look at line 10 of the listing after GOSUB 9000 you will find a little FOR/NEXT loop with READ and DATA. These little two lines sets up a UDG character which is a pointing finger cursor used in the program. By using such as a guideline, you can set up READ and DATA statements with the codes this program comes up with to do lots of User Defined Graphics characters much more simply in your own programs. A word of caution: the cursor takes up position "u" (or CHR$ 164), so don't try to overwrite it.

Good luck in your designing efforts. If you have any comments or improvements, feel free to send them to me. Also, if you are not up to the task of typing in a long listing, then a copy with refinements, bells and whistles is available on a new cassette tape for $5. Send to: Paul Bingham, P.O. Box 2034, Mesa, AZ 85204.

```
  1 REM

 "ULTRA-EASY DESIGNER GRAPHICS"
   version 1.0 by p. bingham

  10 GO SUB 9000: FOR t=65528 TO
 65535: READ o: POKE t,o: NEXT t
: DATA 0,0,63,252,252,248,0,0
  20 DIM d(32): DIM u(20): LET c
s=7: LET at=7: LET px=5: LET py=
1: INPUT "Press ENTER to continu
e...";n$: GO TO 105
  21 LET x=1: LET y=5: GO TO 400
  22 LET x=1: LET y=13: GO TO 40
0
  23 LET x=9: LET y=5: GO TO 400
  24 LET x=9: LET y=13: GO TO 40
0
  25 FOR m=21 TO 24: GO SUB m: N
EXT m: RETURN
  26 PAPER 7: LET k=5+(8*INT (px
/13)): FOR h=k TO k+7: PRINT AT
py,h;CHR$ 128: NEXT h: RETURN
  27 PAPER 0: LET k=5+(8*INT (px
/13)): FOR h=k TO k+7: PRINT AT
py,h;CHR$ 143: NEXT h: RETURN
  28 INPUT "1)SAVE UDGs  2)LOAD
UDGs  ";t: IF t=1 THEN SAVE "UDG
"CODE 65368,159: RETURN
  29 LOAD "UDG"CODE 65368,159: G
O SUB 9000: RETURN
  30 LET s=1: LET x=1: LET y=4:
LET q=1: GO TO 35
  31 LET s=9: LET x=1: LET y=12:
 LET q=22: GO TO 35
  32 LET s=17: LET x=9: LET y=4:
 LET q=1: GO TO 35
  33 LET s=25: LET x=9: LET y=12
: LET q=22: GO TO 35
  34 FOR m=30 TO 33: GO SUB m: N
EXT m: RETURN
  35 GO SUB 300: RETURN
  36 LET s=1: GO TO 48
  37 LET s=9: GO TO 48
  38 LET s=17: GO TO 48
  39 LET s=25: GO TO 48
  40 FOR m=36 TO 39: GO SUB m: L
ET f=1: NEXT m: RETURN
  41 INPUT "1)COPY  2)DISPLAY  "
;t: IF t=1 THEN COPY : LPRINT :
LPRINT : LPRINT : RETURN
  43 DIM t$(4): INPUT "Which in
12347(EX:abct):";t$: FOR t=1 TO
4: GO SUB 350: LET h=((CODE t$(t
))-97)*8+65368: FOR m=h TO h+7:
LET a1=PEEK m
  44 FOR g=8 TO 1 STEP -1: LET a
1=a1/2: IF INT a1<a1 THEN PAPER
0: PRINT AT y1,x1+g;CHR$ 143;: L
ET a1=INT a1: GO TO 46
  45 PAPER 7: PRINT AT y1,x1+g;C
HR$ 128;
  46 NEXT g: LET y1=y1+1: NEXT m
: NEXT t: RETURN
  48 INPUT "UDG letter as Storag
e:";v$: IF CODE v$>65 AND CODE v
$<123 THEN GO TO 55
  49 INPUT "Illegal entry! ENTER
 continues ";v$: GO TO 48
  50 LET q=1
  55 IF u(q)<>0 THEN LET w=u(q):
 GO TO 480
  57 IF q=20 THEN LET w=1: GO TO
480
  60 LET q=q+1: GO TO 55
  65 IF CODE v$>96 AND CODE v$<1
17 THEN LET w=CODE v$-96: GO TO
480
  70 IF VAL v$>0 AND VAL v$<21 T
HEN LET w=VAL v$: GO TO 480
  75 GO TO 48
 100 GO SUB 9000
 105 PAPER 1: PRINT AT cs,26;" "
; PAPER 7: PRINT AT cs,26;CHR$ 1
64: GO SUB 8000: PAPER 1: PRINT
AT cs,26;" "
 110 IF CODE INKEY$=51 THEN LET
cs=cs+1: GO TO 120
 115 GO TO 140
 125 IF cs=9 THEN LET cs=10
 130 IF cs=15 THEN LET cs=16
 135 IF cs>21 THEN LET cs=1
 137 PAPER 7: PRINT AT cs,26;CHR
$ 164
 140 IF CODE INKEY$=52 THEN LET
cs=cs-1: GO TO 160
 150 GO TO 190
 165 IF cs=9 THEN LET cs=8
 170 IF cs=15 THEN LET cs=14
 175 IF cs<1 THEN LET cs=21
 180 PAPER 7: PRINT AT cs,26;CHR
$ 164
 190 IF CODE INKEY$=49 THEN GO S
UB cs+20
 200 GO TO 105
 300 FOR h=x TO x+7: DIM a(8): L
ET b=1: LET c=0
 310 FOR t=8 TO 1 STEP -1: LET a
(t)=INT ((ATTR (h,y+t))/8)
 320 IF a(t)<>7 THEN LET c=c+b
 330 LET b=b*2: NEXT t: LET d(s)
=c: LET s=s+1
 340 PAPER 5: BRIGHT 1: PRINT AT
 h,q;"  ": PRINT AT h,q;c: NEXT
 h: BRIGHT 0: RETURN
 350 IF t<=2 THEN LET y1=1: LET
x1=(INT (t*6/12))*8+4: RETURN
 360 LET y1=9: LET x1=(INT (t*3/
12))*8+4: RETURN
 400 PAPER 7: FOR h=x TO x+7: FO
R t=y TO y+7: PRINT AT h,t;CHR$
128: NEXT t: NEXT h: RETURN
 480 LET j=INT (s/8): LET i=6536
0+8*w: LET e=w+143: PAPER 7
 490 FOR t=i TO i+7: POKE t,d(s)
: LET s=s+1: NEXT t
 500 LET ki=i: LET f=e: FOR h=18
+j TO 21: PRINT AT h,12;CHR$ (f-
79);AT h,14;ki;AT h,20;f;AT h,24
;CHR$ f: LET ki=ki+8: LET f=f+1:
 NEXT h
 510 PRINT AT 18,j+1;CHR$ (e-79)
: FOR h=19 TO 21: PRINT AT h,j+1
;CHR$ e: NEXT h
 515 IF j<2 THEN PRINT AT 19,j+7
;CHR$ e: RETURN
 520 PRINT AT 20,j+5;CHR$ e: RET
URN
8000 GO SUB 8100
8002 LET n=CODE INKEY$: IF n=53
THEN GO SUB 8100: LET px=px-1
8005 IF px<5 THEN LET px=20
8010 IF n=54 THEN GO SUB 8100: L
ET py=py+1
8015 IF py>16 THEN LET py=1
8020 IF n=55 THEN GO SUB 8100: L
ET py=py-1
8025 IF py<1 THEN LET py=16
8030 IF n=56 THEN GO SUB 8100: L
ET px=px+1
8035 IF px>20 THEN LET px=5
8040 IF n=48 THEN PAPER 7: PRINT
 AT py,px;CHR$ 128
8042 IF n=57 THEN PAPER 0: PRINT
 AT py,px;CHR$ 143: PAPER 7
8045 FOR t=103 TO 105: PLOT 104,
t: PLOT t,104: NEXT t
8050 RETURN
8100 LET at=INT ((ATTR (py,px))/
8)
8105 IF at=7 THEN PRINT AT py,px
;CHR$ 134: PRINT AT py,px;CHR$ 1
37: PRINT AT py,px;CHR$ 128
8110 IF at=0 THEN PAPER 7: PRINT
 AT py,px;CHR$ 134: PRINT AT py,
px;CHR$ 137: PAPER 0: PRINT AT p
y,px;CHR$ 143: PAPER 7
8980 RETURN
9010 BORDER 1: BRIGHT 1
9015 PAPER 5: BRIGHT 1: FOR t=0
TO 21: PRINT AT t,0;"
                     ";: NEXT t
9020 PAPER 1: BRIGHT 0: FOR t=0
TO 21: PRINT AT t,26;" ": NEXT t
9030 PAPER 5: BRIGHT 1
9050 PAPER 1: BRIGHT 1: FOR t=0
TO 21: PRINT AT t,27;"      ": NE
XT t
9060 PAPER 5: BRIGHT 0: FOR t=4
TO 21: PRINT AT 0,t;" ";AT 17,t;
" ": NEXT t
9070 FOR t=1 TO 16: PRINT AT t,4
;" ";AT t,21;" ": NEXT t
9080 PRINT AT 0,4;"_";AT 0,21;"_
";AT 8,4;" ";AT 8,21;"_"
9090 FOR t=32 TO 39: PLOT 39,t:
PLOT 103,t: PLOT 168,t: PLOT 39,
t+136: PLOT 103,t+136: PLOT 168,
t+136: NEXT t
9092 PRINT AT 0,8;"-1-";AT 17,8;
"-3-";AT 0,15;"-2-";AT 17,15;"-4
-"
9093 PLOT 39,39: DRAW 0,129: DRA
W 129,0: DRAW 0,-129: PLOT 32,39
: DRAW 143,0
9095 PAPER 1: INK 7: PRINT AT 0,
27;"ERASE";AT 7,27;"SWATH";AT 8,
27;"SAVE ";AT 9,27;"CODES";AT 15
,27;"STORE";AT 21,27;"PRINT"
9100 BRIGHT 1: LET a=28: LET j=1
: GO SUB 9500
9105 LET j=10: GO SUB 9500
9110 LET j=16: GO SUB 9500
9120 PRINT AT 6,a;"row"
9130 BRIGHT 0: PAPER 7: FOR t=1
TO 16: PRINT AT t,5;"
      ": NEXT t
9200 INK 0: PAPER 5: BRIGHT 1
9215 PRINT AT 18,1;"ABCD"
9220 PRINT AT 18,11;"'A'";AT 19,
11;"'B'";AT 20,11;"'C'";AT 21,11
;"'D'"
9225 PAPER 7: BRIGHT 0: PRINT AT
 19,1;"     ";AT 20,1;"     ";AT 2
1,1;"     ";AT 19,7;"  ";AT 20,7;
"  "
9250 RETURN
9500 PRINT AT j,a;"-1-";AT j+1,a
;"-2-";AT j+2,a;"-3-";AT j+3,a;"
-4-";AT j+4,a;"all": RETURN
```

POLY-SCROLL

by S.D. Lemke

POLY-SCROLL demonstrates a screen utility for a T/S 2068 Demo used at the T/S Computerfest In Ohio.

PS uses the alternate display file (Display File 2) to present pictures or data to the observer while it is preparing the next screen on the standard display file (Display File 1). Reference Appendix C: Display Modes and Memory of the 2068 User Manual. Using the alternate display file instead of the standard display file has several advantages: the computer can be writing on the screen (DF-1) with the usual BASIC commands, while you look at DF-2. Your screen will not be affected by what is going on with DF-1 until you scroll the data to DF-2. You can LOAD SCREEN$ or other data and not have your picture interrupted with the BASIC LOADer messages printed on DF-1. You are in direct control of what the observer can see (or not see).

The program is mostly BASIC, with 3 short machine code routines--Col, Row, and Vid (as used in the program listing). These routines are POKEd into memory locations starting from a user defined value, BASE (line 60). The machine code is totally relocatable using the LOADer (lines 8000 to 8140). The LOADer has a built in checker to validate the code as it is POKEd into memory. If there is an error, a warning will be given and the line with the error will be identified. The present Demo uses a BASE value of 48500. If you are not using the 2040 type printer, the printer buffer location (23296) would work very well.

COL and ROW each transfer a column or row of picture element data from display file 1 to display file 2.

Each is very fast. Each of these routines use locations 23727 and 23728 to define the row and column respectively, that is to be transferred. Rows are defined as 1 (bottom) to 24 (top), and columns are 2 (right) to 33 (left). The top, left location is row 24, column 33 and the bottom, right is row 1, column 2. The reason for this odd arrangement is that ROW and COL use a ROM call based on this arrangement. By transferring rows and columns from Display File 1 to Display File 2 in different orders, you can scroll data onto the screen from top to bottom, bottom to top, left to right, ect. The BASIC used to do this is found in lines 9100 to 9210. A total of twelve different scrolls are presented!

VID is the machine code that prepares the TS 2068 for the use of the alternate display file. This routine relocates the Function Dispatcher and Machine Stack.

After you type in the program, it will self-save by typing RUN 9999 [ENTER]. To run, just RUN [ENTER]. The program will demonstrate all 12 screen scrolls and pause after the last. Press ENTER to start the demo again. Any other key will return you to the standard display file and the program will "STOP".

```
  1 REM ************************

    P O L Y -- S C R O L L

   by      S D Lemke
   Lemke Software Development
       2144  White  Oak
     Wichita,  Ks.   67207

**********************************

  10 REM     BASE = 48500
  20 REM   RND  48525 -- Column
  30 REM   RND  48552 -- Row
  40 REM   RND  48587 -- vid mode
  50 REM ************************

  60 LET base=48500: LET b1=INT
(base/256): LET b2=base-256*b1:
LET base2=base+15: LET b3=INT (b
ase2/256): LET b4=base2-256*b3
  70 LET col=base+25
  80 LET row=base+52
  90 LET vid=base+87
 100 REM ************************

 200 GO SUB 8000: RANDOMIZE USR
vid: OUT 255,0: CLS : GO SUB 910
0: OUT 255,1
 210 PAPER 1: BORDER 1: INK 9: C
LS
 220 PRINT "This program uses th
e alternate Display File to disp
lay data or text. All the usual
Basic print and draw commands ca
n be used.": GO SUB 9100: PAUSE
300
 310 PAPER 2: INK 9: CLS
 320 PRINT '''"Each time you ex
ecute a ROW or  Column Machine C
ode routine, 1  Row or Column is
 copied from    Display File 1 t
o Display File 2": GO SUB 9110:
PAUSE 300
 410 PAPER 3: INK 9: CLS
 420 PRINT AT 6,0;"  While the r
eader is reading      the present
 screen, you can      have the pr
ogram prepare the     next screen
."
 425 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175:
 GO SUB 9120: PAUSE 300
 510 PAPER 4: INK 9: CLS
 520 PRINT AT 10,0;" This provid
es the programmer a  powerful to
ol for creating fast and exiting
 games and demos.": GO SUB 9130:
 PAUSE 300
 610 PAPER 5: INK 9: CLS
 620 PRINT AT 13,0;"This program
 will demonstrate a total of 12
different ways that you can scol
l a screen!": GO SUB 9140: PAUSE
 300
 710 PAPER 6: INK 9: CLS
 720 PRINT AT 15,0;"I hope that
you enjoy using the POLY-SCROLL
screen utility.": GO SUB 9150: P
AUSE 240
 810 PAPER 7: INK 1: CLS
 820 CIRCLE 128,88,40: CIRCLE 12
8,88,41: PRINT AT 10,16;"1": CIR
CLE 110,100,3: CIRCLE 146,100,3:
 CIRCLE 112,99,1: CIRCLE 148,99,
1: PLOT 110,70: DRAW 36,0,.5: GO
 SUB 9160: PAUSE 200
 910 PAPER 0: INK 9: CLS
 920 PRINT AT 1,0;" One advantag
e of using DF-2 to display data
is that you can nowload in data,
 pictures, etc. andnot have your
 display affected  by the LOAD t
itles printed on   DF-1": GO SUB
 9170: PAUSE 300
1010 PAPER 1: INK 9: CLS
1020 PRINT AT 5,0;"After the LOA
D is complete you  can scroll yo
ur data/picture to DF-2 for view
ing!": GO SUB 9180: PAUSE 300
1110 PAPER 2: INK 9: CLS
1120 PRINT AT 8,0;"When customiz
ing this program   for your own
use, delete lines  210 thru 1320
 of this demo.": GO SUB 9190: PA
USE 240
1210 PAPER 3: INK 9: CLS
1220 PRINT AT 11,0;"Set BASE suc
h that it will not  be over writ
ten and destroyed   by your prog
ram. If you are not using the 20
40 printer, then theprinter buff
er location (23296)  is a good v
alue for BASE": GO SUB 9200: PAU
SE 330
1310 PAPER 4: INK 9: CLS
1320 PRINT AT 10,0;"Press ENTER
to Start Demo over  again, any o
ther key to STOP": GO SUB 9210:
PAUSE 0: LET i$=INKEY$: IF i$=CH
R$ 13 THEN  GO TO 210
7998 OUT 255,0: STOP
7999 REM  POKE/LOAD Machine Code

8000 PAPER 1: INK 9: BORDER 1: C
LS : PRINT TAB 11; FLASH 1;"WORK
ING"
8100 LET sum=0: RESTORE 9000: FO
R i=vid TO vid+38: READ a: POKE
i,a: LET sum=sum+a: NEXT i: IF s
um<>6207 THEN  LET l=9000: GO TO
 9800
8110 LET sum=0: RESTORE 9010: FO
R i=base TO base+24: READ a: POK
E i,a: LET sum=sum+a: NEXT i: IF
 sum<>2787 THEN  LET l=9010: GO
TO 9800
8120 LET sum=0: RESTORE 9020: FO
R i=col TO col+26: READ a: POKE
i,a: LET sum=sum+a: NEXT i: IF s
um<>(2694+b1+b2+b3+b4) THEN  LET
 l=9020: GO TO 9800
8130 LET sum=0: RESTORE 9030: FO
R i=row TO row+34: READ a: POKE
i,a: LET sum=sum+a: NEXT i: IF s
um<>(3347+b1+b2+b3+b4) THEN  LET
 l=9030: GO TO 9800
8140 RETURN
9000 DATA 46,0,62,1,211,244,219,
255,203,255,211,255,62,6,245,251
,205,142,14,0,219,255,203,191,21
1,255,175,211,244,241,254,128,32
,4,50,91,104,251,201
9010 DATA 33,176,92,78,33,175,92
,70,120,205,41,9,229,193,201,124
,15,15,15,230,3,246,88,103,201
9020 DATA 205,b2,b1,6,192,229,20
5,b4,b3,126,17,0,32,25,119,225,1
26,25,119,17,224,31,237,82,16,23
5,201
9030 DATA 205,b2,b1,6,8,197,6,32
,229,229,205,b4,b3,126,17,0,32,2
5,119,225,126,25,119,225,35,16,2
37,193,17,224,0,25,16,227,201

9098 REM  Scroll Subroutines

9099 REM    Columns   ==>>
9100 PRINT AT 20,3;"Left to Righ
t --- No. 1": FOR c=33 TO 2 STEP
 -1: POKE 23727,24: POKE 23728,c
: RANDOMIZE USR col: NEXT c: RET
URN
9109 REM    Columns   <<==
9110 PRINT AT 20,3;"Right to Lef
t --- No. 2": FOR c=2 TO 33: POK
E 23727,24: POKE 23728,c: RANDOM
IZE USR col: NEXT c: RETURN
9119 REM    Rows      Upward
9120 PRINT AT 20,2;"Bottom to To
p --- No. 3": FOR r=1 TO 24: POK
E 23727,r: POKE 23728,33: RANDOM
IZE USR row: NEXT r: RETURN
9129 REM    Rows      Downward
9130 PRINT AT 20,2;"Top to Botto
m --- No. 4": FOR r=24 TO 1 STEP
 -1: POKE 23727,r: POKE 23728,33
: RANDOMIZE USR row: NEXT r: RET
URN
9139 REM    Diag    L>R  T>B
9140 PRINT AT 20,0;"Diagonal: L>
R and T>B --- No. 5": LET c=33:
FOR r=24 TO 1 STEP -1: POKE 2372
7,r: POKE 23728,33: RANDOMIZE US
R row
9142 IF c>=2 THEN  POKE 23727,24
: POKE 23728,c: RANDOMIZE USR co
l: LET c=c-1
9144 IF c>=2 AND r>=17 THEN  POK
E 23727,24: POKE 23728,c: RANDOM
IZE USR col: LET c=c-1
9146 NEXT r: RETURN
9149 REM    Diag    R>L  T>B
9150 PRINT AT 20,0;"Diagonal: R>
L and T>B --- No. 6": LET c=2: F
OR r=24 TO 1 STEP -1: POKE 23727
,r: POKE 23728,33: RANDOMIZE USR
 row
9152 IF c<=33 THEN  POKE 23727,2
4: POKE 23728,c: RANDOMIZE USR c
ol: LET c=c+1
9154 IF c<=33 AND r>=17 THEN  PO
KE 23727,24: POKE 23728,c: RANDO
MIZE USR col: LET c=c+1
9156 NEXT r: RETURN
9159 REM    Diag    R>L  B>T
9160 PRINT AT 20,0;"Diagonal: R>
L and B>T --- No. 7": LET r=1: F
OR c=2 TO 33: POKE 23727,24: POK
E 23728,c: RANDOMIZE USR col
9162 IF r>=17 THEN  POKE 23727,2
4: POKE 23728,c+1: RANDOMIZE USR
 col: LET c=c+1
9164 IF r<=24 THEN  POKE 23727,r
: POKE 23728,33: RANDOMIZE USR r
ow: LET r=r+1
9166 NEXT c: RETURN
9169 REM    Diag    L>R  B>T
9170 PRINT AT 20,0;"Diagonal: L>
R and B>T --- No. 8": LET r=1: F
OR c=33 TO 2 STEP -1: POKE 23727
,24: POKE 23728,c: RANDOMIZE USR
 col
9172 IF r>=17 THEN  POKE 23727,2
4: POKE 23728,c-1: RANDOMIZE USR
 col: LET c=c-1
9174 IF r<=24 THEN  POKE 23727,r
: POKE 23728,33: RANDOMIZE USR r
ow: LET r=r+1
9176 NEXT c: RETURN
```


Listing continued next page...

```
9179 REM      Around - Inward
9180 PRINT AT 20,0;"Around:  Out
side In --- No. 9": LET r=2: LET
 l=33: LET t=24: LET b=1: FOR x=
1 TO 16: POKE 23727,24: POKE 237
28,1: RANDOMIZE USR col: LET l=1
-1
9182 POKE 23727,t: POKE 23728,33
: RANDOMIZE USR row: LET t=t-1
9184 POKE 23727,24: POKE 23728,r
: RANDOMIZE USR col: LET r=r+1
9186 POKE 23727,b: POKE 23728,33
: RANDOMIZE USR row: LET b=b+1
9188 NEXT x: RETURN
9189 REM      Around - Outward
9190 PRINT AT 20,0;"Inside to Ou
tside --- No. 10": LET r=18: LET
 l=17: LET t=17: LET b=8: FOR x=
1 TO 16: POKE 23727,24: POKE 237
28,1: RANDOMIZE USR col: LET l=1
-1
9192 POKE 23727,t: POKE 23728,33
: RANDOMIZE USR row: LET t=t-1
9194 POKE 23727,24: POKE 23728,r
: RANDOMIZE USR col: LET r=r+1
9196 POKE 23727,b: POKE 23728,33
: RANDOMIZE USR row: LET b=b+1
9198 NEXT x: RETURN
9199 REM     Columns    => <=
9200 PRINT AT 20,0;"Left and Rig
ht to Center --- 11": FOR c=33 T
O 2 STEP -1: POKE 23727,24: POKE
 23728,c: RANDOMIZE USR col: POK
E 23727,24: POKE 23728,2+33-c: R
ANDOMIZE USR col: NEXT c: RETURN
9209 REM     Rows       Up/Down
9210 PRINT AT 20,0;"Top and Bott
om to Center --- 12": FOR r=1 TO
 24: POKE 23727,r: POKE 23728,33
: RANDOMIZE USR row: POKE 23727,
25-r: POKE 23728,33: RANDOMIZE U
SR row: NEXT r: RETURN
9800 CLS : PRINT "You have an er
ror in line ";l: BEEP 1,1: STOP
9998 STOP
9999 SAVE "Poly-Scrol" LINE 1
```

T/S V.I.P. *Timex/Sinclair Video Identification Program*

by Bill Ferrebee

A few issues ago (Vol.2 No.1), I presented "LABEL MAKER", a program to print cassette labels for your program collection. The response I received showed me that many of you found this program a useful one.

I now present for your approval: T/S V.I.P. (Timex/Sinclair Video Identification Program).

T/S V.I.P. allows you to use standard 3 1/2" X 15/16" tractor-feed labels with your full size printer to organize your video collection.

The labels are printed in the format as shown in the example provided.

As with "Label Maker", this program is designed to work with the print driver software provided with your particular printer interface. The procedure is simple:

1. Load your driver software, and save it to a blank tape. DO NOT REWIND THE TAPE!

2. Enter the program listing. Save this on the blank tape immediately following the driver. Use the command: SAVE "vip" LINE 1

That's it! You should be able to find tractor-feed labels wherever you buy computer paper. Good luck, and let me know of any enchancements!

```
Inventory #
Title
Rating
Length
Format                    Speed
```

```
10 INK 0: CLS : INPUT "Inventory #:";a$
12 IF LEN a$>32 THEN  GO TO 10
14 PRINT TAB 16-(LEN a$/2);a$
20 INPUT "Title:";b$
22 IF LEN b$>32 THEN  GO TO 20
24 PRINT TAB 16-(LEN b$/2);b$
30 INPUT "Rating:";c$
32 IF LEN c$>32 THEN  GO TO 30
34 PRINT TAB 16-(LEN c$/2);c$
40 INPUT "Length (ex. 1:23):";d$
42 IF LEN d$>32 THEN  GO TO 40
44 PRINT TAB 16-(LEN d$/2);d$
50 INPUT "Format:";e$
52 IF LEN e$>15 THEN  GO TO 50
54 PRINT e$;
60 INPUT "Speed:";f$
62 IF LEN f$>15 THEN  GO TO 60
64 PRINT TAB 32-LEN f$;f$
70 PRINT AT 21,8; FLASH 1;" Correct? (y/n) "
72 IF INKEY$="" THEN  GO TO 72
74 IF INKEY$="n" THEN  GO TO 10
 80 PRINT AT 21,8;"                      "
 90 INPUT "Number of copies:";x
100 FOR i=1 TO x
110 LPRINT TAB 16-(LEN a$/2);a$
120 LPRINT TAB 16-(LEN b$/2);b$
130 LPRINT TAB 16-(LEN c$/2);c$
140 LPRINT TAB 16-(LEN d$/2);d$
150 LPRINT e$;TAB 32-LEN f$;f$
160 LPRINT : NEXT i
170 INPUT "More? (y/n):";x$
172 IF x$="y" THEN  GO TO 90
180 INPUT "Another title? (y/n):";x$
182 IF x$="y" THEN  GO TO 10
190 CLS : PRINT AT 10,8;"[Work  Complete]": STOP
```

DATAGEN

by Kenneth Fracchia

Have you ever typed a long list of information into an array, and then attempted to RUN your program, only to find out that the RUN command wiped out the entire array? The first thing you say to yourself (after cursing at the computer) is "why didn't I put the data in a Data Statement?"

DATA Statements have some other advatages over arrays. When storing strings, all the strings in an array must be as long as the longest string (shorter strings are followed by spaces). Therefore, if you're trying to conserve bytes, a data statement may be the answer. You can also save RAM bytes by storing numbers as strings, and then using the VAL function when you need the numbers for mathematical calculations.

Now, why don't we use DATA Statements more often? The answer is simple: it takes forever to type in all those quotation marks, commas, and line numbers. To eliminate this problem, I wrote a program that allows you to type in your data, depressing only the ENTER key between each item in your list. To use this program, we must first reserve room in RAM for the DATA Statement. To do this, first MERGE any other program which has more bytes of BASIC than your longest DATA Statement will have. If you do not have a program which is long enough, you can MERGE several programs. Do not be concerned that the MERGEd programs do not make sense. They are only used to occupy space in RAM. It's better to MERGE a program(s) that is much longer than necessary, otherwise, a long DATA Statement might overflow the reserved bytes of RAM. 20,000 bytes is probably sufficient for your longest DATA Statement, and does not take too long to MERGE.

Now, MERGE "DATAGEN", and then enter the following command: GOTO 9999. This will SAVE all the merged programs, including "DATAGEN". Now, you can either LOAD "DATAGEN", or RUN 9976. The following should be printed before the program listing. CAUTION: RUNing this program will delete some (or all) of the program, so SAVE it immediately after typing it into your computer. The command GOTO 9999 will save it.

MORE PROGRAM NOTES: Be aware that if you input a high number (say 20000) as the desired DATA Statement length, then if you do not enter 20000 bytes of data, there will be a delay as the program pokes in spaces from the last entered piece of data to the end of the line. Example: You ask for 20000 bytes in the DATA Statement, and only enter 1000 bytes...then it may take about 10 minutes to POKE 19000 spaces into the data line. This problem will not occur in normal use of the program.

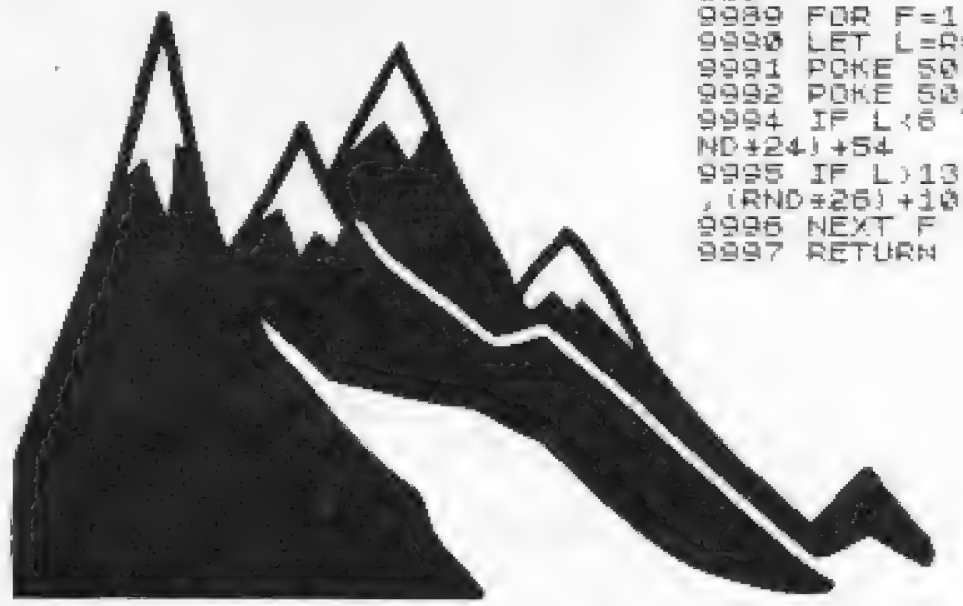
```
9972 REM   MERGE a program(s)
with more bytes of BASIC than
your longest DATA STATEMENT.
9973 REM   MERGE "DATAGEN"
9974 REM   GOTO 9999 to SAVE
the above merged programs.
9975 REM   Now you can either
LOAD "DATAGEN" or RUN 9976
9976 CLS : POKE 23561,200: POKE
23609,100: INPUT "How many bytes
 do you want in   the DATA state
ment?  ";n
9977 INPUT "What will be the lin
e number    of the DATA statemen
t?"''"(the line number must be le
ss   than 9997)";l  IF NOT (l<99
97) THEN GO TO 9977
9978 PRINT INVERSE 1;AT 10,10,"P
LEASE WAITE": LET prog=PEEK 2363
5+256*PEEK 23636
9979 LET adr=prog-1
9980 LET adr=adr+4+PEEK (adr+3)+
256*PEEK (adr+4)
9981 IF prog+5+n<adr THEN LET ad
r=adr: GO TO 9983
9982 GO TO 9980
9983 POKE prog,0  POKE prog+1,1
9984 LET bytes=ad-prog-3: POKE p
rog+3,INT (bytes/256): POKE prog
+2,bytes-256*INT (bytes/256)
9985 POKE prog+4,228
9986 CLS : PRINT "Enter data,  D
epress the ENTER  key after each
 entry."''"Enter the word ""end""
 to end the list"''"Number of b
ytes in"''"DATA list: ";bytes-3;
"(this includes quotes, commas,"
''"and the word ""DATA"").
9987 LET adr=prog+4
9988 LET adr=adr+1: POKE adr,34
9989 BEEP .02,1  BEEP .02,1: INP
UT d$: IF d$="end" OR d$="END" T
HEN CLS : GO TO 9994
9990 IF adr+LEN d$+4>ad THEN CLS
 : PRINT AT 5,0;"""";d$;""";" d
oes not fit in this"''"data list.
  Start another"''"data list.": G
O TO 9994
9991 FOR z=1 TO LEN d$: LET adr=
adr+1: POKE adr,CODE d$(z)  NEXT
 z
9992 LET adr=adr+1: POKE adr,34
 LET adr=adr+1: POKE adr,44
9993 GO TO 9988
9994 PRINT AT 10,10;"PLEASE WAIT
E": FOR z=adr-1 TO ad-1: POKE z,
32: NEXT z
9995 BEEP 1,5: BEEP 1,-5: BEEP 1
,5: BEEP 1,-5: PRINT AT 15,0;"De
press letter D to DELETE this pr
ogram.  The DATA LIST will notbe
 deleted."
9996 IF INKEY$<>"d" AND INKEY$<>
"D" THEN GO TO 9996
9997 DELETE 2,9996: POKE prog,IN
T (l/256): POKE prog+1,l-256*INT
 (l/256): CLS
9998 DELETE 9997,
9999 SAVE "DATAGEN" LINE 9976
```

SKI

by
Charles E. Goyette

```
  1 LET IN=1: POKE 23658,8: BOR
DER 5: SOUND 6,31;7,63-24;8,16;9
 10;12,14: LET HI=0: CLS : PRINT
 FLASH 1;AT 10,9;"STOP THE TAPE"
  PAUSE 150: CLS : INPUT "DO YOU
 WANT TO SAVE ?(Y-N)",A$  IF A$=
"Y" THEN SAVE "SKI" LINE 1
  2 GO SUB 700: PAUSE 240: CLS
  GO SUB 1000
  3 LET H=0
  4 IF IN=1 THEN GO SUB 4000
  5 LET ROLE=1: INPUT "HOW MANY
 PLAYERS ?";TOT: GO SUB 500
  6 PRINT AT 21,0; FLASH 1;"
    WAIT 10 SECONDS
GO SUB 9980
  7 LET SUB=0: LET S$=""
  8 REM CORRESPONDANCE:  A=4 ,
B=k , AB=A , C=/ , D=- , E=\ ,
DE=/\ , F=▶
          G=| , H=\ , I=/
  9 CLS
 10 LET Z=50000: LET A=15
 11 LET H=0: LET A$="|"
 12 POKE 23692,3: PRINT AT 15,0
,"PRESS ON KEYS 5 OR 8 TO START.
","/////////"
 13 PRINT AT 21,8;"GET READY ";
Y$(ROLE)
 14 BEEP .07,10: PAUSE 0
 15 CLS
 17 LET TRE=0
 20 LET SL=0
 25 LET T=8
 30 PRINT AT 0,A;T;" ENERGY"
 40 SOUND 6,31;7,63-24;8,16;9;1
0;12,14
 45 POKE 23692,255
 49 IF A=0 THEN LET A=31
 50 IF A=31 THEN LET A=0
 51 IF INKEY$="5" THEN LET A=-1
: SOUND 13,4;1;10: LET A$="/"
 52 IF INKEY$="8" THEN LET A=1
: SOUND 13,4;1;10: LET A$="\"
 53 LET A=A+H
 54 IF PEEK (22848+A)<>56 THEN
GO TO 300
 55 IF PEEK (22849+A)<>56 THEN
GO SUB 325
 59 PRINT AT 10,A;A$
 70 PRINT AT 21,0
 75 PRINT TAB (PEEK (Z)); INK 4
;"k"
 77 IF PEEK (Z)>50 THEN PRINT A
T 20,0: PRINT TAB (PEEK (Z))-50;
 INK 5;"/^\"
 79 IF PEEK (Z)>100 THEN PRINT
AT 20,0: PRINT TAB (PEEK (Z))-10
0; INK 2;"▶   ▶": PRINT TAB (PEE
K (Z))-100; INK 2;"|____|"
 80 LET Z=Z+1
 90 IF Z>50210 THEN LET Z=Z+50:
GO TO 400
 100 GO TO 45
 110 LET Z=Z-50000
 120 LET Z=Z+SL
 123 CLS
 125 IF TRE=0 THEN LET Z=Z+50  P
RINT AT 1,0; INK 2;"YOU HAVE HIT
 NO TREE!!BONUS:50!"
 155 PAUSE 50
 157 PRINT AT 3,0
 160 PRINT Y$(ROLE);" YOU HAVE "
;Z;" POINTS"
 162 PRINT
 165 IF Z<SUB THEN PRINT S$;" HA
S THE BEST SCORE:",SUB
 171 PRINT
 173 IF Z>=SUB THEN IF TOT>1 THE
N LET SUB=Z: LET S$=Y$(ROLE): PR
INT S$;" HAS THE BEST SCORE:";SU
B
 174 PRINT
```

```
 175 IF Z>=HI THEN LET HI=Z: LET
 H$=Y$(ROLE)
 176 PRINT H$;" HAS THE RECORD:"
;HI
 177 PAUSE 50
 180 LET ROLE=ROLE+1
 185 IF ROLE>TOT THEN GO TO 500
 215 GO TO 10
 300 IF PEEK (22848+A)=60 THEN L
ET T=T-4: LET TRE=TRE+1: SOUND 6
,31;7,7;8,16;9,16;10,16;12,30;13
,9: FOR 0=1 TO 70: NEXT 0: SOUND
 6,31;7,63-24;8,16;9,10;12,14
 310 IF PEEK (22848+A)=61 THEN S
OUND 7,56;8,15;9,15;10,15;11,0;1
2,0: FOR 0=1 TO 250 STEP 20: SOU
ND 0,0;2,250-0;4,0: NEXT 0: LET
 SL=SL+5: LET L=INT (RND*8)-4: LE
T A=A+L: SOUND 6,31;7,63-24;8,16
;9,10;12,14;13,9;10,16
 315 IF PEEK (22848+A)=58 AND SC
REEN$ (10,A)="_" THEN FOR F=1 TO
 5: BEEP .008,10: NEXT F: LET T=
T+5
 317 PRINT AT 9,A;T
 318 IF T<=0 THEN GO TO 400
 320 GO TO 59
 325 IF PEEK (22849+A)=60 THEN L
ET T=T-4: LET TRE=TRE+1: SOUND 6
,31;7,7;8,16;9,16;10,16;12,30;13
,9: FOR 0=1 TO 70: NEXT 0: SOUND
 6,31;7,63-24;8,16;9,10;12,14
 330 IF PEEK (22849+A)=61 THEN S
OUND 6,0;7,56;8,15;9,15;10,15;11
,0;12,0: FOR 0=0 TO 250 STEP 20:
 SOUND 0,0;2,250-0;4,0: NEXT 0:
LET SL=SL+5: LET L=INT (RND*8)-4
: LET A=A+L: SOUND 6,31;7,63-24;
8,16;9,10;12,14;13,9;10,16
 332 IF PEEK (22849+A)=58 AND SC
REEN$ (10,A)="_" THEN FOR F=1 TO
 5: BEEP .008,10: NEXT F: LET T=
T+5
 334 PRINT AT 9,A;T
 335 IF T<=0 THEN GO TO 400
 336 RETURN
 400 FOR F=8 TO 21: PRINT AT F,0
;"  ";: NEXT F
 401 FOR F=1 TO 21 STEP 2
 403 FOR G=1 TO 8: NEXT G
 404 PRINT AT 21,0; INK 4;"▲ ▲
▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ "
 406 NEXT F
 410 GO TO 110
 500 PAUSE 100: IF TOT>1 THEN PR
INT AT 10,0;H$;" HAS WON";AT 11,
10;"WITH ";SUB;" POINTS.",,,,,,,
,,,,,,,,
 505 FOR U=1 TO 5
 510 BEEP .03,0: BEEP .03,3: BEE
P .03,7: BEEP .03,1: BEEP .03,4:
 BEEP .03,9: BEEP .1,15
 515 NEXT U
 517 LET ROLE=1
 520 FOR F=1 TO 50: NEXT F
 525 INPUT "DO YOU PLAY ANOTHER
GAME ?(Y-N)",S$: IF S$<>"N" THEN
 GO TO 550
 530 RESTORE : GO TO 2
 600 DIM Y$(TOT,9)
 605 FOR F=1 TO TOT
 610 PRINT AT 21,0;"NAME OF PLAY
ER ";F;: INPUT U$
 611 LET U$=U$+"          "
 615 LET Y$(F)=U$(1 TO 9)
 620 NEXT F
 630 RETURN
 650 IF TOT=1 AND Z>150 THEN GO
TO 6
 651 IF TOT=1 THEN GO TO 7
 652 LET I$=Y$(1)
 655 FOR F=1 TO TOT-1
 660 LET Y$(F)=Y$(F+1)
 665 NEXT F
 670 LET Y$(TOT)=I$
 675 IF SUB>150 THEN GO TO 6
 699 GO TO 7
 700 FOR F=0 TO 7
 705 READ X,Y
 710 POKE USR "A"+F,X
 715 POKE USR "B"+F,Y
 720 NEXT F
 730 FOR F=0 TO 7
 733 READ M,N,O
 735 POKE USR "C"+F,M
 740 POKE USR "D"+F,N
 742 POKE USR "E"+F,O
 750 NEXT F
 755 FOR F=0 TO 7
 760 READ M,N,O,P
 765 POKE USR "F"+F,M: POKE USR
"G"+F,N
 767 POKE USR "H"+F,O: POKE USR
"I"+F,P
 770 NEXT F
 800 RETURN
1000 PLOT 95,150
1010 DRAW -50,-40: DRAW 50,-20:
DRAW -50,-40
1020 DRAW 55,40: DRAW -50,20: DR
AW 45,40
1030 PLOT 180,150
1040 DRAW 4,-100: DRAW 4,100: DR
AW -4,-7: DRAW -4,7
1050 CIRCLE 184,157,7
1060 PLOT 120,150
1070 DRAW 0,-100: DRAW 4,47: DRA
W 35,-47: DRAW -35,52: DRAW 35,4
7: DRAW -35,-52: DRAW -4,47
1080 PAUSE 140
1100 PRINT AT 1,5;"▲";AT 16,3;"
▲";AT 20,17;"▲";AT 8,27;"▲";A
T 4,25;"▲";AT 17,22;"▲";AT 10,
5;"▲";AT 5,10;"▲";AT 2,15;"▲"
1200 FOR F=1 TO 20: BEEP .002,65
: NEXT F: PRINT AT 18,1;"JUNE '8
6-CHARLES ERIC GOYETTE"
1205 PAUSE 15
1210 FOR F=1 TO 40: BEEP .002,65
: NEXT F: PAUSE 80: BEEP .04,5:
BEEP .02,10: BEEP .05,5
1230 PAUSE 40
2000 RETURN
4000 LET IN=IN+1: CLS
4010 PRINT : PRINT "************
*RULES***************"
4020 PRINT : PRINT "YOU START TH
E GAME WITH 8 UNITS OF ENERGY .
WHEN YOU HIT A TREE,"
4030 PRINT "YOU LOSE 4 UNITS . W
HEN YOU SKI BETWEEN TWO FLAGS,YO
U GET 5     UNITS OF ENERGY ."
4040 PRINT "WHEN YOU STEP ON A /
\ , YOU GETS BONUS POINTS . DES
CENDING ALL THE SLOPE GIVES 50 B
ONUS POINTS."
4045 PRINT "IF YOU HIT NO TREE Y
OU GET 50    BONUS POINTS."
4047 PRINT "WHEN ONE OF THE PLAY
ERS GETS     MORE THAN 150 POINTS
,A NEW SLOPEIS CREATED."
4050 PRINT : PRINT "***********G
OOD LUCK!***********"
4060 RETURN
9000 DATA 0,128,1,128,7,192,25,1
52,33,132,15,200,49,140,65,130
9010 DATA 0,0,0,0,0,0,0,126,0,3,
129,192,12,0,48,48,0,6,192,0,3,0
,0,0
9020 DATA 48,32,128,1,56,32,64,2
,62,32,32,4,63,32,16,8,62,32,8,1
6,56,32,4,32,48,32,2,64,32,32,1,
128
9989 FOR F=1 TO 200 STEP 2
9990 LET L=RND*15
9991 POKE 50000+F,(RND*28)+1
9992 POKE 50001+F,(RND*28)+1
9994 IF L<6 THEN POKE 50000+F,(R
ND*24)+54
9995 IF L>13.5 THEN POKE 50000+F
,(RND*26)+101
9996 NEXT F
9997 RETURN
```



TASWORD Word Count Utility Modification by Duncan Teague

A word count utility is a feature I've wanted to add to my copy of Tasword Two for nearly a year. The routine published in the February/March 1986 issue of ZX COMPUTING [a British magazine], is wonderful! Thanks to John Wall's article, my already excellent version of Tasword has been made even more useful.

I want to share with you a small modification that is necessary to allow this Spectrum utility to work correctly with the U.S. version of Tasword Two.

The values in the data statement will not work on the Timex/Sinclair 2068 version of Tasword because of the different starting address of the text file code and the maximum length of the text file. In the U.S. version of Tasword, the text file begins at address 33280 instead of 32000. The maximum length of the text file is only 19200 bytes instead of 20480 - 300 lines instead of 320.

Because of these differences, two values must be altered in the data statement in line 9810. The sixth set of three digits must be changed from "081" to "076". The ninth set of three digits must be changed from "124" to "129".

The latter alteration raises the starting address of the word count from 31999 to 33279. This effectively reduces the maximum length of the text file by 1280 bytes. The former alteration allows the word count to proceed through only 19200 + 256 bytes of text file instead of 20480 + 256 bytes.

Here's the complete routine as modified for the Timex/Sinclair 2068:

```
9800 RESTORE : READ a$: FOR n=52
610 TO 52657: LET a=VAL a$( TO 3
): POKE n,a: LET a$=a$(4 TO ): N
EXT n
9810 DATA "00100000001700076033
25512903502706200018620012625403
20402450030350270620001862000620
63165254000032007035126254032043
03223512625403203223102421 7"
```

I have underlined the two critical three-character decimal numbers to be sliced from this string. The user still must modify the program's menu to add a PRINT USR VAL "52610" at some appropriate point and modify the "Save Tasword" routine to start saving the Tasword code at 52610 for a length of 12925 bytes.

For reference to this converted Spectrum program, take a look at the original article in ZX COMPUTING.

MacIntosh-type Menu for the 2068

by Dennis Jurries

Several months ago, I wrote a weld design program that used two special menus from which a person could select the number of the weld picture that would be most applicable. Figure number one is a print-out of one of these menus (the actual screen display is much more impressionable).

For this project, I also developed special graph paper for high-resolution drawings, on my Radio Shack Plotter. With this graph paper, and the step by step approach in this article, you can make your very own "MacInclair" menus. [NOTE: TIME DESIGNS will mail a copy of this graph paper, suitable for photo-copying, to any reader who sends a legal-size S.A.S.E., to: Time Designs Magazine, Graph Paper Offer, 29722 Hult Rd., Colton, OR 97017. The graph paper is free. One copy per customer please.]

First draw a border "grid" (see example) on the graph paper to enable you to keep your pictures in your menu separate from each other. Draw or photo-copy your pictures, and glue them onto the separate grid areas. Write a subroutine such as the one I used (see below) in my weld program to set up the grid on the screen.

```
5000 REM Grid Pattern
5010 DRAW 255,0 : DRAW 0,152 : DRAW -255,0 :
DRAW 0,-152 : PLOT 1,1 : DRAW 253,0 : DRAW
0,150 : DRAW -253,0 : DRAW 0,-150 : PLOT 2,
115 : DRAW 251,0 : DRAW 0,1 : DRAW -251,0 :
PLOT 2,75 : DRAW 251,0 : DRAW 0,1 : DRAW -25
1,0 : PLOT 2,35 : DRAW 251,0 : DRAW 0,1 : DR
AW -251,0 : PLOT 63,2 : DRAW 0,148 : DRAW 1,
0 : DRAW 0,-148 : PLOT 127,2 : DRAW 0,148 :
DRAW 1,0 : DRAW 0,-148 : PLOT 191,2 : DRAW 0
,148 : DRAW 1,0 : DRAW 0,-148
5020 RETURN
```

Next we will write a subroutine that will enable us to draw the picture from data that we put into a data statement.

```
6000 REM Draw Subroutine
6001 LET z=0: LET i=18: LET j=133
6005 READ a: IF a=VAL "3000" THEN GO TO VAL
"6009"
6006 IF a=VAL "1000" THEN READ a: READ b: RE
AD c: DRAW a,b,c: GO TO VAL "6005"
6007 IF a=VAL "2000" THEN READ a: READ b: PL
OT a,b: GO TO VAL "6005"
6008 READ b: DRAW a,b: GO TO VAL "6005"
6009 LET z=z+1: IF z=1 OR z=2 THEN GO TO VAL
"6005"
6010 STOP
6050 CLS
```

Cut and paste the flat plates that are butt welded together onto the graph paper so that the lower left corner of the picture is at 18 horizontal from left to right and 133 from the bottom (refer to the coordinates given on the graph paper). Write a statement similar to the grid pattern statement that describes only the picture (not the arrows, as in my pictures). Do not use plot statements, but redraw over lines to make a "continuous" picture. The graph paper helps to determine the length and direction of the draw statements. Now convert your statement to a DATA statement similar to line 8010 in the subroutine below. Now do the same for the arrows and any portion of the weld that may be required, and compare with line 8000 below. Next we will write a subroutine to blow the picture up into a larger size, to check for any errors. Note that "i" and "j" are the original picture plot points, and the numbers are the added locations for the start of the arrows and welds in line 8000.

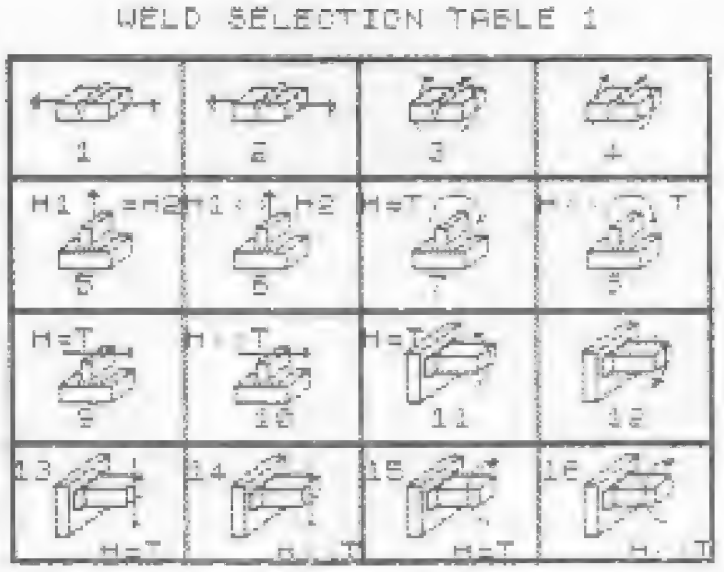


Figure 1

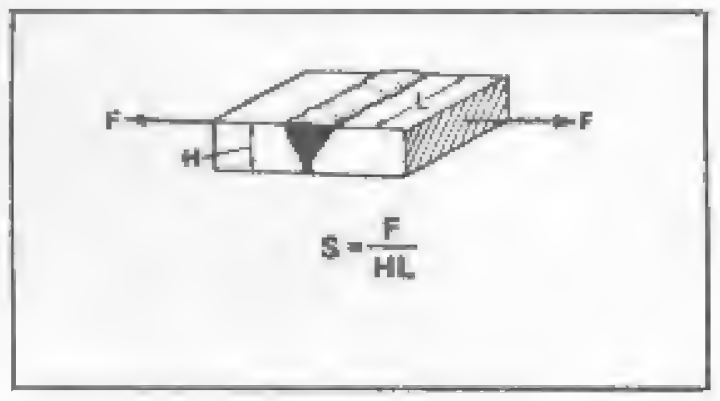


Figure 2

```
7000 REM Expanded Draw Sub.
7005 LET t=1: LET i=50: LET j=100: PLOT i,j
7010 RESTORE 8000+t*10
7020 READ a : IF a=3000 THEN GOTO 7050
7030 IF a=1000 THEN READ a : READ b : READ c
: DRAW a,b,c : GOTO 7020
7035 IF a=2000 THEN READ a : READ b : GOTO 7
020
7040 READ b : DRAW 3*a,3*b : GOTO 7020
7050 LET t=t+1: IF t=2 THEN GO TO 6005
7060 STOP
8000 DATA 2000,i+1,j+6,-10,0,2,2,0,-5,-2,2,2
000,i+23,j+5,14,0,-2,2,0,-5,2,2,2000,i+9,j,2
,5,-2,-5,-2,5,2000,i,j,3000
8010 DATA 19,0,0,5,0,5,-7,0,-0,-5,0,5,1000,-
4,0,.5*PI,-0,-5,0,5,-0,0,-0,-5,0,-5,19,0,0,5
,-7,0,1000,-4,0,.5*PI,-0,0,0,0,1000,4,0,-.5*
PI,7,0,0,5,-0,-5,0,-5,3000
```

By pressing CONTINUE after the first RUN, and the addition of another DATA statement such as line 8020, the arrows and the weld portion that was left out can be added to a picture that is three times the original size.

```
8020 DATA 2000,54,110,-25,0,0,4,-0,-4,0,-4,0
,4,2000,122,110,30,0,0,4,0,-4,-0,-4,0,4,2000
80,100,6,15,-6,-15,-6,15,3000
```

The original grid pattern may also be run through the subroutine at line 6000.

Using this information as a background, you can now make "MacInclair"-type menus for your programs and drawings. Just add numbers to each grid, and a statement at the bottom of the screen to prompt the user to select a number. No fancy Koala tablet, no Mouse, and no joystick necessary.

DK' Tronics Sound Synthesizer

Reviewed by Duncan Teague

DK'Tronics Sound Synthesizer
Damco Enterprises
67 Bradley Court
Fall River, MA 02720
Tel. (617) 678-2110
For Spectrum or 2068 (with
Rainbow Plus Interface)
Price: $39.95

The DK'Tronics Sound Synthesizer is a toy, a tool, and a terrific add-on to your Spectrum or T/S 2068. As a toy it lets you enjoy the three cannel sound your computer is capable of generating, and enjoy it at more impressive levels. As a tool it helps programmers and/or musicians compose music and develop sound effects for program enhancement or pure enjoyment. The Synthesizer plugs directly into the back of your Spectrum computer or can be used with T/S 2068 computers equipped with the Rainbow Plus Spectrum Emulator/Interface. With the Sound Synthesizer comes a four-inch speaker and one meter of cable for connecting the two. The volume level of the speaker can be controlled by a knob on the interface, or from within the software which accompanies it.

The plug at the end of the speaker cable is an ordinary 1/8-inch (miniature) audio connector. With an appropriate adapter cable you could, I suppose, connect the Systhesizer to the input of your stereo amplifier. Then you could play your computer sound through your hi-fi system.

With the Sound Synthesizer in place, Spectrum software will play its music and sound effects through the speaker. The beeps from your keyboard will be amplified as well. The volume is controlled by the knob on the Synthesizer. (You should hear my Spectrum "Pinball" game.) The peripheral does not operate with a 2068 in its "home" mode...only in Spectrum mode.

The software which accompanies the DK'Tronics Sound Synthesizer is even more impressive than the hardware. "Sound Designer" is capable of generating, storing, and playing back "tones" in three part harmony. It can also produce "white noise" sound effects by modifying the characteristics of the sound produced. The three tone and noise "voices" can be turned on and off, and their volumes can be controlled from within the program.

For keyboard-synthesizer buffs, the Sound Designer software gives you an envelope generator capable of producing square, triangle, and sawtooth wave shapes. You can also control the attack and decay times. There is no provision for controlling the sustain and release.

For music buffs, there is a provision to "play" a five octave music keyboard at the bottom of the screen. Either computer keyboard controls or a joystick can be used to move an indicator to different notes and store them as one of the three "tunes" in memory. Up to 256 notes can be "recorded" per voice and 768 notes per tune. Playback "tempo" is also controllable.

Tune number one has already been stored on the tape. It's a familiar English melody recorded by Simon and Garfunkel in their rendition of "Parsley, Sage, Rosemary and Thyme". Be sure to remove your socks before playing this tune! If you don't, the Sound Synthesizer will surely knock them off. It's simply beautiful.

I played a couple of instruments as a youth, so I'm sufficiently familiar with music to record some simple tunes. In about half an hour (I'm not a keyboard player) I recorded tune number two, "Twinkle, Twinkle Little Star", in three part harmony. An eighth grade fellow T/S enthusiast across the street from me recorded tune number three, Beethhoven's "Fur Elise", in about ten minutes. (Show-off!)

The instuction booklet that comes with the package is slim but informative. It contains some "OUTput" routines for producing music and sound from BASIC. The booklet also covers coarse/fine tuning for notes of the scale, ranges of values for envelope shapes and periods, and instructions for "Sound Designer".

The DK'Tronics Sound Synthesizer is a sound (pun intended) value. The hardware works well in concert with the software and with other programs. This combination will render speechless the Commodore users who brag about the music their computers will produce. You'll also be the envy of your next users group meeting.

SOFTWARE IN REVIEW

Saboteur!

RATING: * * * *

A recent trend in Spectrum programs brings the action and realism of the Martial Arts to the monitor screen. Such programs as "The Way of the Exploding Fists", "Yie Ar Kung Fu", the popular new "Way of the Tiger", and "Saboteur!". There are some folks that just don't appreciate this sort of thing. (Could it possibly be the violent kicks, punches, use of weapons, and resulting in the complete annihilation of your opponent?) Well, perhaps the taste for this has to be acquired like a fine wine. And I'll be the first to admit that I can't pass up a good Bruce Lee or Chuck Norris movie on the telly. It's make believe and all just good fun.

My personal favorite is SABOTEUR! from Durell Software. Although the program is for the Spectrum, it is my understanding that Knighted Computers here in the States has converted this program for the T/S 2068, and uses the joystick port. If you have the Spectrum version, you can still load it in the 2068 using a Spectrum Emulator or Romswitch.

Why SABOTEUR! works so well, is it's state-of-the-art graphics, and that the principal character in this game/simulation is a "Ninja". Yes, perhaps the most ruthless and mystical "style" of the Martial Arts. And this Ninja is no exception...he has all the tricks of the trade including access to shirkens, swords, several explosives, and of course a kicking/punch combination.

TIME DESIGNS SOFTWARE RATING SYSTEM

*	Buy at your own risk.
* *	Mediocre.
* * *	A diamond in the rough.
* * * *	Try it, you'll like it.
* * * * *	Receives "Hall of Fame" status.

As a Ninja and special government agent, your mission is to approach a huge warehouse by sea, search the building for a special computer disk, and escape with it via a helecopter that is parked on the roof. In the meantime, you must thwart off armed guards (who also are trained in the "arts"), guard dogs and automatic laser weapons that track you by video camera. Its a very difficult mission indeed. Fortunately, there a nine skill levels to choose from. Level one will allow you to practice what you need to know for other more difficult levels.

The Ninja is controlled with five keys on the keyboard, or by one of the popular British joystick interfaces for this Spectrum version. Your Ninja can climb up ladders, jump, kick up and out, climb down or crouch (to avoid being kicked and punched), move right or left, take objects, and throw or use these objects.

The animation in SABOTEUR! is just suberb. It must be seen to be appreciated. Along with a complex series of rooms, ladders, and platforms, it all adds up to be a truly enjoyable and addictive game (a bit frustrating at first). Probably not everyones "cup of tea".

SABOTEUR! is available from several British software houses for £8.95. Contact Knighted Computers in the U.S. for the special T/S 2068 version.

--D. Hutchinson

Colonize The Universe

RATING: * * * *

As Commander of your interplanetary spacecraft, there are certain risks and weighted decisions upon your shoulders. Should you send the probecraft into the ominous Black Hole, only to return without the precious fuel and having spent 10 gallons itself? Or should you go on...hoping to reach your own planet Armedia, which you successfully colonized just last week? Your Science Officer predicts a meteor storm is moving into the area.

Taking your second option, you reach the sub-tropic outpost in space by morning. And just in time, as both fuel and food supplies have dangerously diminished. Here on Armedia, you will restock the ship, as this is a prosperous community. You are approached by the Mayor.

"Greetings Master! Welcome Home. How may we serve you?" inquires your humble subject, the Mayor of this colony.

All in a days work, you are heading out once again into the voids of space to search out new planets, where precious fuel factories and food harvesting operations may be installed. Along the way, it is certain that "deals" will be made, Time Warps will be encoutered, and Super Novas could blow your sleek spacecraft into tiny fragments at any time. Which planet will you land on tomorrow...Cyg X-1A, Delcior, or maybe Triffid?

Welcome to the game, COLONIZE THE UNIVERSE, and the delightful imagination of Timothy Kessler, the author of this new program for the Timex/Sinclair 2068. COLONIZE THE UNIVERSE is part text adventure game, part Monopoly-type game, and greatly reminiscent of a boxed board-type game (popular in college circles). The game requires at least 2 players, but will accomodate up to four players.

The graphics portion of COLONIZE THE UNIVERSE is a single screen, the game board itself, which is unique in design. It is shaped somewhat like an upside-down "U". Game players are represented by arrows that flicker when "ready", and circle the parameter of the board depending upon the roll of the dice. The "dice" are two little icons just below the playing area. Text is also continuously displayed below the game board, along with prompts.

The object of the game is "survival of the fittest" with the last player still moving about the game board, taking the title of "winner". Please note, that a single game lasts for hours...so be prepared to spend some time at it.

COLONIZE THE UNIVERSE is a good conceptual game, and I strongly recommend it to afficiando's of this game type. There is no "alien-shooting" here, only strategy and chance. It was refreshing to play a quiet (no sound effects in this one) but challenging round with my computer friends. Mr. Kessler's occaisonal dash of humor, greatly enhanced the flavor of this space saga.

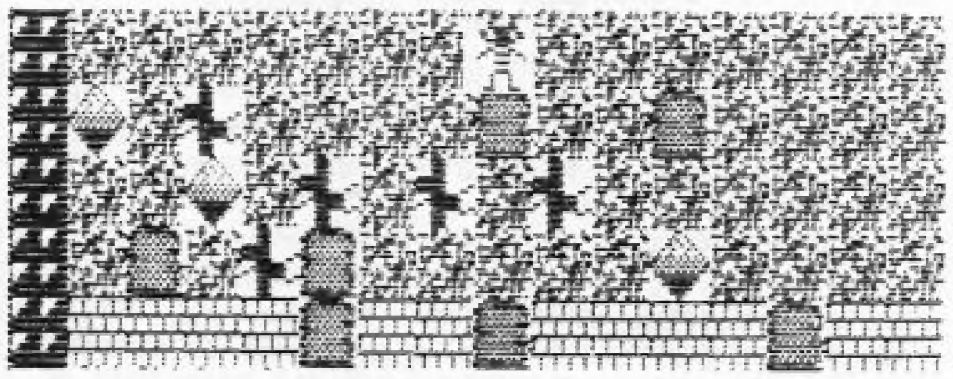
While traveling through space, taking the part of universal real-estate agent, be sure to watch your vitals: fuel, food and currency (gold). And also those "sexy fem-aliens" may rob you blind.

COLONIZE THE UNIVERSE is exclusively distributed by WMJ Data Systems, 4 Butterfly Drive, Hauppauge, New York 11788. Price: $16.95 (+ $3 total order S&H).

--D. Hutchinson

DIAMOND MIKE II

NEW MACHINE CODE ARCADE HIT!



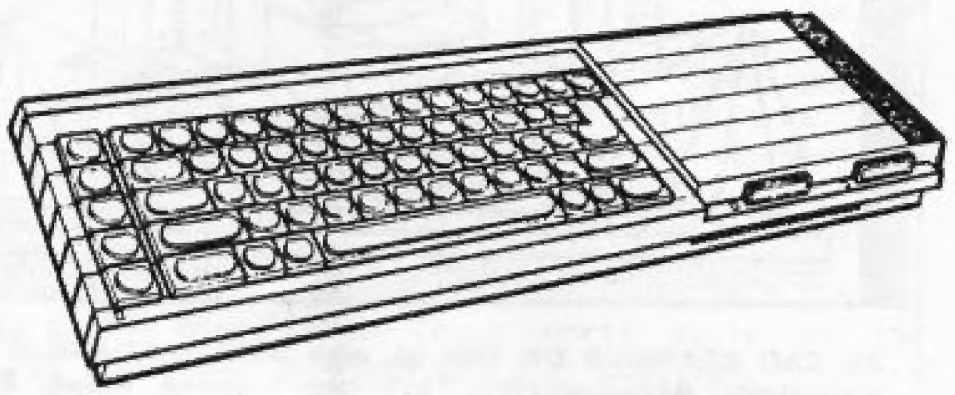
JRC SOFTWARE proudly announces a breakthrough in 2068/Spectrum compatible games called DIAMOND MIKE II. It is a true arcade quality game with brilliant graphics, color and sound. The object is to collect enough diamonds before time runs out, while avoiding falling rocks, ferocious amebas and killer butterflies. There are 22 different screens and 6 levels! DIAMOND MIKE II is sale priced at $17.95! Ten day money-back guarantee! Just $2.00 extra for C.O.D. Or mail check or money order to:

JRC SOFTWARE
P.O. Box 448
Scottsburg, IN 47170
Phone (812) 752-5106 or 752-6071
Diamond Mike II is 100% 16K Machine Code.
Free "Electronic Catalog" with your order!

C. W. Associates

419 N. Johnson Street

Ada, Ohio 45810



*** AUTHORIZED QL DEALER ***

For information and prices:
Send legal-size SASE
or
call (419) 634-4874 (6:00pm-9:00pm)

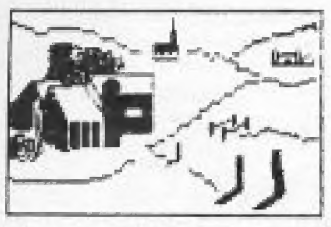
LARKEN DISK DRIVE

Controller Boards for **ZX-81** **2068**

Features Common to Both:
- LDOS on Eprom supports Basic Programs, Code blocks and Arrays
- Commands are-DIRECTORY,LOAD,SAVE,FORMAT, DELETE and EXIT
- Up to 52 files per disk - Directory is DOS maintained
- Can be modified to work on either computer (eprom and cable req)
- The most Powerful DOS for the ZX-81 : Very User Friendly
- 2068 version is Spectrum Emulator Compatible: Extra commands
- Requires double sided 5.25" drive :60 day money back guarantee

Prices : $95.00 (US) for single drive . $99.00 for 1 or2 drives
$ 4.00 for drive cable (all other cables included)
include $5.00 shipping : Send certified cheque or M.O.
To - LARKEN ELECTRONICS
(specify ZX-81 or 2068) RR#2 NAVAN ONTARIO
CANADA K4B-1H9

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$            THE               $
$         KRUNCHER             $
$            IS                $
$        ! !HERE! !            $
$                              $
$    HAVE YOU EVER RUN INTO    $
$    THAT BRICK WALL CALLED    $
$       "OUT OF MEMORY"?       $
$                              $
$ HAVE WE GOT THE ANSWER FOR   $
$  YOU! A PROGRAM THAT WILL    $
$    "BYTE" INTO YOUR BASIC    $
$  PROGRAM, CHEW ON IT FOR A   $
$  FEW SECONDS, AND SPIT IT    $
$  BACK OUT WITH FROM 10 TO    $
$  40% OF THE MEMORY EMPTY!    $
$                              $
$ SOUND GOOD! SURE IT DOES!!   $
$ WE HAVE TESTED IT, AND IN    $
$ ONE CASE, WE WENT FROM 239   $
$ BYTES FREE, TO 12400 FREE!   $
$                              $
$  2068/SPECTRUM - $9.95+1     $
$  1000 VERSION COMING SOON!   $
$                              $
$       JUST CALL OR WRITE:    $
$                              $
$            RMG               $
$     ENTERPRISES              $
$      1419 1/2 7TH STREET     $
$     OREGON CITY, OR 97045    $
$         (503) 655-7484       $
$           S.A.S.E.           $
$       FOR FREE CATALOG       $
$ SEND THIS AD OR COPY WITH    $
$ ORDER-GET COUPON GOOD FOR    $
$ $5 OFF NEXT ORDER FOR $25    #
$           OR MORE!           $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```



Great New Graphics Design Program for the 2066 by

Lemke Software Development

Pixel Sketch and Graphics Editor 2068 version 2.0 is an update of the original PS/GE 2068 with the same powerful graphic editor capabilities: WINDOW COPY, ERASE, ROTATE, ZOOM, WIDE VIEW, PAINT SHADE/FILL, MIRROR IMAGE, INVERT, and DIGITIZE. MERGE parts of two screens into one screen. Full use of all TS 2068 attributes.

Pixel Sketch and Graphics Editor 2068 uses the ADVANCED VIDEO MODES for EXTENDED COLOR and HI RESOLUTION (512 x 176) GRAPHICS. PS/GE 2068 version 2.0 is JOYSTICK controlled, MENU driven, EASY to use, and FUN. FREEHAND draw and erase are 2 times faster now. "OOPS!" function helps un-do errors. Compete 20⁸ page user guide. NEW FUNCTIONS: CONNECT will connect designated points together. CHANGE FONT activates STANDARD, BOLD, MODERN, and *ITALICS* set modifiers. Any character set can be modified to get BOLD, MODERN or ITALICS versions of that character set. LOAD FONT allows the user to LOAD their own favorite FONT. *A sample font - CHANCERY - is provided with the program. With CHANGE FONT you get BOLD CHANCERY.* Pixel Sketch and Graphics Editor 2068 version 2.0 is Tasman, Aerco, and A&J Mikrodrive CPI compatible in al of the 2068 video modes (Standard, Extended Color, and Hi Resolution). Compatible with most dot matrix printers. Can access variable dot density functions of STAR and EPSON type printers to control page size and location! Also 2040 type printer compatible. Get YOUR copy of: *Pixel Sketch and Graphics Editor 2068 version 2.0* from LEMKE SOFTWARE DEVELOPMENT today! Only $19.95 ppd. (owners of the original PS/GE can send original tape (proof-of-purchase) and pay only $10.95). Send legal size SASE for information and a complete catatlog of top quality programs. EXTRA: Now with your order you get FREE a 32 column screen to 64 column (hi resolution) screen conversion utility program. Can even combine 2 standard screens to make 1 hi resolution screen!!

Lemke Software Development

S D Lemke
Lemke Software Development
2144 White Oak
Wichita, KS 67207

Deep in the land of the Hidden Kingdoms, guarded by a vicious and evil dragon, can be found the Elixir of Life...

Dragonhold

DRAGONHOLD is an animated graphic adventure game for the Sinclair QL
And just LOOK at these features -

- Four-way SCROLLING map graphics (equivalent to over 250 screens!)
- Individual computer controlled animated characters that you can communicate with
- Extremely detailed hi-res graphics
- Both visual and text puzzles
- Game variations and new challenges each time a new game is generated
- Save game feature - you can save and play as many different variations as you like!

Available on microdrive cartridge for **$29.95** including p. & p. Send cheque or P.O. to

English Micro Connection
15 Kilburn Court — Newport, RI 02840
401/849-3805

***** SMART TEXT TS-2068 *****

Administrative Software

Data Base - Word Processing
Mailing List - Repeat Printing
Form Letters - Auto Letter Head
Auto Sign Off- Labels/Envelopes
All integrated functions.

Versions For:
AERCO DISK - OLIGER DISK
CASSETTE - ZEBRA OS-64 Cart

Each Package includes four programs. The Cassette and Oliger Disk versions includes one for the OS-64 Cart.

The most comprehensive software for the TS-2068.

$34.00 post paid- Check or MO
Bill Jones, Gulf Micro
1317 Stratford Ave,
Panama City, FL 32404
Inquiries welcome

PRICE BREAKTHROUGH!

Thanks to a **MAJOR** cost reduction, we can now offer the Rotronics **WAFADRIVE** (Less **Rainbow** Interface) for **ONLY $99.95!** The **WAFADRIVE** offers ● ***TWO*** 128K high speed drives operating at almost 2K per second ● An RS232 (serial) and a Centronics (parallel) port that allow **WAFADRIVE** to run almost any full size printer ● Extended Basic operating system ● A blank wafer ● and the **Spectral Writer** word processor program.

The **WAFADRIVE** is a **Spectrum** compatible device, and when used with the **Rainbow Plus** interface, your Timex 2068 will run thousands of **Spectrum** programs.

The **Rainbow Plus Spectrum** interface is both a spectrum emulator AND a spectrum hardware adaptor all in one compact case. For only **$49.95** we will even throw in free Spectrum software to get you started!

TO ORDER:
Send Check, Money Order
VISA or MASTERCARD to:

DAMCO ENTERPRISES
67 Bradley Ct.
Fall River, MA 02720

or call **(617) 678-2110**

CLASSIFIED

FREE ADS FOR SUBSCRIBERS

FOR T/S 2068: 3-Reel 9-Way Win Giant Slot Machine on cassette. All in BASIC & listable, about 32K with Stick/Inkeys provision. Send $6.00 to D. JEWETT, 63 Charro Dr., Santa Rosa, CA 95401. Includes shipping.

FOR SALE: I have an extra copy of MTerm II for TS-2068. Brand new, never used. List for $29.95, will sell for $25.00 ppd. Chia-Chi Chao, 73 Sullivan Dr., Moraga, CA 94556.

FOR SALE: T/S 2068; AERCO FD-68; 2-DS/DD 5¼ Drives; Drive Case & Power Supply; Aerco & Tasman CPI; A&J Micro Drive; All Cables, manuals & S/W. Also CP/M Disc; War In The East; PRO/FILE; Taswide; Tasword; Tech Draw Jr.; MSCRIPT; Fish Locator & others. Most on Disc & Tape.
---EVERYTHING FOR $500---
N. G. Robins, Rt. 3 Box 744, Hayes, VA 23072.

MONEY MACHINE. Exciting new word game for the T/S 2068. May be habit forming. From ABBA Soft. Tape $10; Llist $3, ppd. Herb Bowers, 2588 Woodshire Circle, Chesapeake, VA 23323.

WANTED: A&J Micro Drive, Disk Drive for 2068 & other hardware. No T/S 1000 wanted. Send list with your lowest price. E. Salter, 402 Pine Terrace Cir., Milton, FL 32570, (904) 623-8178.

SOFTWARE for the Timex 2068. Send a S.A.S.E. for a free catalog to: TIMEWARE, 1907 1/2 West Genesee St., Syracuse, NY 13204.

WANTED: T/S 2068 User's Manual in good condition. Stuart Walton, PO Box 513, Rowley, MA 01969.

16K ZX81, Keyboard, 8K BBU Memory, some software & books, $100. TS2068, 2 cart. prg. Hot-Z, Tech Manual & books, $100. F. Schubert RD#3 Box 19, Selkirk, NY 12158.

FOR SALE: Atari 800XL, Disk, Printer Interface, DOSXL, BASICXL, HomePack, 13 books and 33 magazines. All for $275.00. Call (512) 992-6297.

WANTED: Pokes for MSCRIPT, TW+64, TASWORD II, AERCO print driver... to drive AK ROYAL Lettermaster Printer. TS2068, Aerco Interface. Fee. C. Grosh, 1112 Chautauqua Blvd., Pacif. Palis., CA 90272.

T/S 2068 PROGRAMS. MARQUEE: your message crawls across the screen BANNER: prints big bold message on 2040 in 16 sizes. Both programs on tape $10. Brant Wilcox, Box 8, Cawker City, KS 67430.

FOR SALE: 14 issues SYNC, 6 TS Users, 2 SQ mag., 4 K-Power, 6 Synchro-sette, 23 Syntax in hard binder, 55 issues total for $30 postage paid, call (614) 846-3543.

FOR SALE: 2 non-working WC2050 modem boards. $5 each or best offer. Brian Higgins, RR 1, Prescott, KS 66767.

WANT TO BUY: the Timex/Sinclair 2068 graphic color book titled Timex/Sinclair Color Graphics, written by N. Hampshire. Send info to David Britt, 3995 Eloise, Bmt., TX 77707.

2068 SOFTWARE and HARDWARE: Koala Pad with Zebra Painter. Protex Word Processor, Flight Simulator, Diamond Mike. Much more. Reasonable. Send S.A.S.E. to Bob Smith, 428 Marsh Ave., Reno, NV 89509 for complete list.

FOR SALE-TS2068. Used only one month. Includes cables, 3 software cassettes. $100 including shipping anywhere in USA. John Gerdes, 123 N. Front St., Lewisburg, PA 17837.

Do you have some equipment or a program that you would like to sell? Looking for something hard to find? Place an ad in **THE CLASSIFIEDS!** Subscribers can place one free personal ad in each issue. Ad size is 32 Col. wide (like 2040 paper) and maximum of six lines. For additional lines - $3 each. **NON-SUBSCRIBERS** and **DEALERS:** $4 a line. **DEADLINE FOR ALL CLASSIFIED ADS:** Two weeks before publication date. Mail your ad to: **TIME DESIGNS MAGAZINE, The Classifieds Dept., 29722 Hult Rd., Colton, Oregon 97017.**

NEW PRODUCTS ANNOUNCEMENT

For TS2068 and SPECTRUM

NovelSoft *PROUDLY PRESENTS THE ALL NEW...*

Artworx

VERSION 1.1

- Pull-Down Menus
- Several Brushes
- Spray
- Auto-Fill
- Zoom
- Undo
- Several Text Fonts
- Cut & Paste Windows
- Auto-Speed Control
- Magnify & Reduce
- Rotate & Mirror
- Full Attribute Control
- Fully Elastic Shapes including Circle, Box Triangle, Ray and Line
- Fast Ellipse and Arc
- Includes Spectrum & TS2068 Versions
- Supports Microdrivers and Kempston Joystick
- Includes GALLERY, the slide show/animator
- 5 Samples of Artwork
- Excellent Manual

Unshackle your creativity with ... ARTWORX!!

ARTWORX V1.1 - $19.95 U.S. plus $3.00 S & H

NovelSoft *INTRODUCES THE REVOLUTIONARY NEW BASIC COMPILER...*

TIMACHINE

The dream of every BASIC programmer has now been realized!

* TIMACHINE will turn your BASIC into super-fast machine code, running up to 200 times faster!
* Handles floating point operations like SIN, COS, TAN
* Handles all BASIC except I/O
* Includes an excellent manual and 4 demonstration programs
* Compiles up to 27K in seconds
* Includes Spectrum & 2068 versions

Super-Charge your BASIC programs with...*TIMACHINE!*

TIMACHINE — $19.95 U.S. Plus $3.00 S & H

NovelSoft
A FORMAT FOR THE FUTURE

106 Seventh Street, Toronto, Ontario, Canada M8V 3B4 • TEL. (416)259-8682 • CompuServe 70416,1435